

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Ing. Vojtěch Dluhý



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

HODNOCENÍ ÚČINNOSTI KLASIFIKACE POMOCÍ ROC KŘIVEK

EVALUATION OF CLASSIFICATION EFFICIENCY USING ROC CURVES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ing. Vojtěch Dluhý

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. Jitka Poměnková, Ph.D.

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Audio inženýrství**
Ústav telekomunikací

Student: Ing. Vojtěch Dluhý

ID: 134470

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Hodnocení účinnosti klasifikace pomocí ROC křivek

POKYNY PRO VYPRACOVÁNÍ:

JB2T: Cílem semestrálního projektu je posoudit účinnost klasifikace zvukového signálu sirén (např. vozidel IZS) v rušném uličním prostředí prostřednictvím ROC křivek. Pro analýzu zvukového signálu využijte odhadu spektrální reprezentace vstupního signálu a předem stanovené hodnoty intenzity zvukové sirény. Doplňkově proveďte posouzení ROC křivky pomocí výpočtu AUC plochy.

JBCT: Proveďte rešerši dostupné literatury na téma klasifikace zvukového signálu pomocí ROC křivek a výpočtu AUC plochy. Vytvořte výpočtovou proceduru pro hodnocení účinnosti klasifikace ROC křivek prostřednictvím metod výpočtu AUC plochy. Aplikujte existující dostupné nástroje na analýzu zvoleného signálu. Simulujte různé způsoby a úrovně zašumění signálu. Zjištěné výsledky porovnejte, vyhodnoťte a formulujte metodická doporučení pro klasifikaci.

DOPORUČENÁ LITERATURA:

[1] WESTIN, L. K. Receiver operating characteristic (ROC) analysis. Evaluating discriminance effects among decision support systems. 28 pages. [online] Cited 2014-02-11. Available at: <http://nutkin.cs.umu.se/research/reports/2001/018/part1.pdf>.

[2] ZOU, G. Y., YUE, L.: Using confidence intervals to compare several correlated areas under the receiver operating characteristic curves, *Statistics in Medicine*, 2013, Vol. 32, pp. 5077-5090

Termín zadání: 1.2.2016

Termín odevzdání: 1.6.2016

Vedoucí práce: doc. RNDr. Jitka Poměnková, Ph.D.

Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce je zaměřena na hodnocení účinnosti detekce pomocí ROC křivek. Seznamuje čtenáře s principem základních algoritmů pro odhad spektra signálu, s výpočtem ROC křivky a získáním hodnoty AUC. Dále obsahuje algoritmu pro detekci sirény v zaznamenaném zvuku při zarušení dpravním prostředím o různých hodnotách SNR, který provádí výpočet ROC křivek a hodnot ploch pod nimi a jejich srovnání. Algoritmus je napsán ve vývojovém prostředí Matlab.

KLÍČOVÁ SLOVA

ROC, AUC, plocha pod křivkou, siréna, šum, rušení, prostředí, Matlab

ABSTRACT

This bachelor thesis is focused on work with the ROC curves. Introduces the reader with ROC curves and their use to analyzing signals and basic principles of spectral estimation and getting an AUC for ROC curve. Also shows an algorithm for detecting sound of siren in record using a simple algorithm and evaluation of various jamming sound of sirens by displaying the ROC curves, supplemented by calculation and comparing the areas under these curves (AUC). The algorithm is written in a development environment Matlab.

KEYWORDS

ROC, receiver operating characteristic, AUC, area under curve, siren, background noise, Matlab

DLUHÝ, Vojtěch *Hodnocení účinnosti klasifikace pomocí ROC křivek*: bakalářská práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 52 s. Vedoucí práce byl doc. RNDr. Jitka Poměnková, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Hodnocení účinnosti klasifikace pomocí ROC křivek“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

BRNO

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucí své bakalářské práce paní doc. RNDr. Jitce Poměnkové, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

BRNO

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

BRNO

.....

podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OP Výzkum a vývoj
pro inovace

OBSAH

Úvod	1
1 Metodika	3
1.1 Odhad kmitočtového spektra	3
1.2 Operační křivky (ROC)	5
1.3 Plocha pod křivkou (AUC)	8
2 Vstupní data	11
3 Aplikace	13
3.1 Konzolová aplikace	13
3.2 Grafické rozhraní	17
4 Výsledky	18
4.1 Siréna s bílým šumem	18
4.2 Sirény s dopravním ruchem	18
4.2.1 První siréna	19
4.2.2 Druhá siréna	20
4.2.3 Třetí siréna	21
4.2.4 Čtvrtá siréna	21
4.2.5 Pátá siréna	22
4.3 Srovnání	23
5 Závěr	24
Literatura	25
Seznam symbolů, veličin a zkratk	28
Seznam příloh	30
A Výpočet ROC	31
B Spektrogramy sirény se šumem	32
C Spektrogramy první sirény	33
D Spektrogramy první sirény s bílým šumem	34
E Spektrogramy druhé sirény	35

F	Spektrogramy třetí sirény	36
G	Spektrogramy čtvrté sirény	37
H	Spektrogramy páté sirény	38
I	Obsah souboru analyza.m	39
J	Obsah souboru roc.m	40
K	Obsah souboru rocvyk.m	41

SEZNAM OBRÁZKŮ

1.1	Spektra částí zvukového souboru.	5
1.2	Tabulka srovnání měřených a skutečných hodnot.	6
1.3	Příklad binormálního rozdělení s prahováním.	6
1.4	Limitní případy ROC křivek.	7
1.5	Výpočet AUC lichoběžníkovou metodou.	9
1.6	Výpočet AUC obdélníkovou metodou.	9
1.7	Srovnání ROC křivek s různým počtem bodů.	10
2.1	Zobrazení časového průběhu a spektragramu.	12
3.1	Vypočítané ROC křivky	14
3.2	ROC křivky zobrazené v jednom grafu, pro lepší srovnání	16
3.3	Grafické rozhraní programu	17
4.1	ROC křivky zvuků sirény s bílým šumem	18
4.2	ROC křivky zvuků první sirény s rušným pozadím	19
4.3	ROC křivky zvuků první sirény s rušným pozadím a bílým šumem	20
4.4	ROC křivky zvuků druhé sirény s rušným pozadím	20
4.5	ROC křivky zvuků třetí sirény s rušným pozadím	21
4.6	ROC křivky zvuků čtvrté sirény s rušným pozadím	22
4.7	ROC křivky zvuků páté sirény s rušným pozadím	22
A.1	Výpočet a vykreslení ROC křivky	31
B.1	Spektrogramy první sirény s přidaným bílým šumem.	32
C.1	Spektrogramy první sirény s dopravním ruchem.	33
D.1	Spektrogramy první sirény s ruchem a bílým šumem.	34
E.1	Spektrogramy druhé sirény s dopravním ruchem.	35
F.1	Spektrogramy třetí sirény s dopravním ruchem.	36
G.1	Spektrogramy čtvrté sirény s dopravním ruchem.	37
H.1	Spektrogramy páté sirény s dopravním ruchem.	38

SEZNAM TABULEK

1.1	Tabulka slovních hodnocení testů dle AUC [22]	10
2.1	Soubory s dopravním ruchem	11
2.2	Soubory jen s bílým šumem	11
4.1	Tabulka srovnání AUC bez bílého šumu a s ním	19
4.2	Tabulka srovnání hodnot AUC všech zvuků	23

ÚVOD

Detekce sirén v dopravním ruchu je při zvyšující se hustotě dopravy problémem zasluhující řešení. Často se při jízdě se zapnutým autorádiem stává, že řidič neslyší okolní ruch mimo svůj vůz, především tedy vozidla Integrovaného záchranného sboru (IZS). Bylo by tedy vhodnou pomůckou zařízení pro zjištění zvuku sirény v okolním prostředí, které by upozornilo řidiče na blížící se vůz např. Rychlé záchranné služby (RZS). Případné zařízení však musí být spolehlivé a pokud možno jednoduché, aby nemohlo dojít selhání obsluhy, tak i samotného zařízení. S výhodou by případný výrobek byl nápomocen i osobám s poruchou sluchu, které by upozornil na blížící se vozidlo se zapnutou sirénou, příp. znějící sirénu obecně.

Operační křivky (ROC z angl. Receiver Operating Characteristic) křivky nabízejí grafické zobrazení vztahu mezi binormálním rozložením, kdy jedna křivka normálního rozložení představuje pravděpodobnost obsažení detekovaného signálu, zatímco druhá představuje pravděpodobnost bez hledaného signálu. Vyhodnocení více ROC křivek se provádí pomocí ploch pod nimi, které svou hodnotou odrážejí jejich tvar. ROC křivky tedy zobrazují schopnost algoritmu přiřazovat určitým objektům jejich vlastnost s ohledem na předchozí znalost zda objekt tuto vlastnost má či nikoli. Uvedme jako příklad testování funkčnosti polovodičových diod, kdy víme zda je daná dioda v pořádku či „proražená“ a zavádíme automatický test, pomocí ROC křivek pak můžeme vyhodnotit účinnost toho testu. Dalším příkladem může být detekce emailových zpráv, zda se jedná či nejedná o spam apod.

Vyhodnocení více ROC křivek se provádí pomocí ploch pod nimi (AUC z angl. Area Under Curve), které svou hodnotou odrážejí jejich tvar. Tvar ROC křivky nám také ukazuje spolehlivost detekce použitého algoritmu, porovnáním ploch pod více křivkami můžeme určit úspěšněji fungující algoritmus, případně lépe detekovatelnější signál (resp. skupinu objektů).

Pro testování navrženého algoritmu pro výpočet ROC a AUC je vybráno několik záznamů policejní sirény bez rušivého pozadí. Tento zvuk je doplněn o několik zvukových pozadí, se kterými je následně analyzován. Jedná se o tři rušná městská či dopravní prostředí a jejich kombinace s bílým šumem, pro ukázkou jeho vlivu na tvar ROC křivek. Algoritmus je doplněn jednoduchým výpočtem AUC, která umožní porovnání několika ROC křivek. Vstupní data jsou detailněji popsána v kapitole 2.

Cílem práce je vyhodnotit účinnost klasifikace pomocí ROC křivek. Ověření schopnosti detekce je provedeno na bance zvuků několika sirén v uličním prostředí. Pro detekci a hodnocení je vytvořen jednoduchý algoritmus pro výpočet ROC křivek a jejich AUC ploch. Tento algoritmus je koncipován jednoduše pro snadnou implementaci do různých zařízení, počínaje jednoduchými mikroprocesory, přes hradlová pole FPGA, počítače s procesory ARM až po výkonné stroje kategorie PC.

Tato práce popisuje princip jednoduché analýzy audio záznamu sirény v rušném uličním prostředí pro získání možnosti vyhodnocení schopnosti algoritmu detekovat zvuk sirény. V kapitole 1 je popsána metodika použitá při vytváření algoritmu, způsoby analýzy, výpočtu ROC křivek a výpočtu AUC. V následující kapitole 2 jsou popsána vstupní zvuková data, jejich získání a použitý datový formát. Další kapitola 3 popisuje provedený experiment využití navrženého algoritmu na vybraném vzorku zvukových nahrávek, následovaná kapitolou 4 s popisem výsledků.

1 METODIKA

Metod pro detekci zvuku sirény existuje více, s úspěchem lze využít například Module Difference Function (MDF)[1] nebo Rychlou Fourierovu transformaci (Fast Fourier Transform – FFT) např. v mikroprocesoru[2]. ROC křivky je možné využít při detekci signálu jako Two-class verifier[3].

MDF využívá pro svou funkčnost detekci intonace určitého kmitočtu. Výstupem je tedy graf, který v čase zobrazuje intonující (čisté) kmitočty. Tento graf je dále analyzován a pokud je po následujících několik vzorků opakující se kmitočet, pak je řečeno, že byla nalezena siréna[1]. Při použití FFT jako v [2] lze detekovat sirénu s podobným výsledkem jako v předchozím případě, tedy s grafem intonujících kmitočtů. Je zde však dvakrát použita FFT, která je často implementována v mikroprocesorech pro signálové zpracování (signal processing). V tomto příspěvku je výsledkem zařízení analyzující zvuk snímaný mikrofonem a signalující přítomnost sirény pomocí LED diody. Při detekci pomocí Two-class verifier[3] (volně přeloženo jako zkoušení na dvě třídy) je vstupní signál se známým stavem (Target (obsahující cílový signál) a Background (obsahující pouze pozadí)) tříděn pomocí trénovacích dat do dvou tříd a následně je zkoumána účinnost detekce, tedy správnost detekce hledaného signálu ve zvuku.

Pro analýzu vybraných zvuků byl vytvořen skript v programovém prostředí Matlab, který využívá následující principy a metody. Kroky analýzy jsou shrnuty v následujících bodech:

- načtení vstupních zvukových záznamů,
- odhad kmitočtového spektra,
- zjištění dominantních prvků spekter (pro stanovení prahu),
- výpočet a vykreslení ROC,
- výpočet AUC.

Hlavní body budou popsány v následujících podkapitolách. Jejich použití je popsáno v kapitole 3.

1.1 Odhad kmitočtového spektra

Načtením zvukových souborů jsou získány vektory jejich dat. Tato data je nutné pro analýzu převést z oblasti časové do oblasti kmitočtové. Pro transformaci je možné využít několika metod, např. metodu periodogramu, Rychlou Fourierovu transformaci (FFT), Yule-Walkerův autoregresní proces nebo metodu Welchova odhadu spektrální hustoty[5].

Meucci používá ve své práci[1] pro odhad spektra MDF, tedy Modulovou diferenční funkci, při které dochází k vyhledávání intonace kmitočtu. Pokud je posléze

v časové oblasti detekováno několik stejných kmitočtů za sebou, je rozhodnuto, že se jedná o zvuk sirény. Miyazaki používá pro odhad kmitočtového spektra FFT. Fourierova transformace je snadno implementovatelná do 16bitového signálového mikroprocesoru, který využívá ve své práci návrhu autonomního zařízení detekce sirény[2]. V senior design projektu pod vedením Bendlera[4] také zmiňují využití FFT v DSP pro detekci zvuku sirény. Výsledkem projektu pak je také autonomní zařízení pro použití v automobilu.

Základní metodou pro odhad spektra je periodogram, vztah pro výpočet je na 1.1.

$$S_x(\omega) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n)e^{-j\omega n} \right|^2, \quad (1.1)$$

kde $n = 0, \dots, N - 1$, N je celkový počet hodnot, $x(n)$ je vstupní signál, n je vzorek signálu a ω je úhlový kmitočet.

Welchova metoda odhadu spektra využívá principu periodogramu. Signál je rozdělen na překrývající se segmenty, které jsou násobeny vybraným oknem. Překrytí segmentů bývá obvykle 50%[5]. Výsledky jednotlivých dílčích spekter se poté průměrují, čímž dostaneme spektrum celkové. Při použití této metody by bylo potřeba pro každý zvukový signál optimalizovat velikost okna a úroveň překrytí, jelikož při užití „optimální“ hodnoty není zaručeno získání správných výsledků pro všechny vstupní signály[6].

Další možností je využít Yule-Walkerovy rovnice pro výpočet parametrů autoregresního procesu. Jedná se o soustavu rovnic (1.2), jejíž rozměr závisí na použitém řádu. Pro výpočet je nezbytné provést optimalizaci řádu pro získání vhodného řádu, který může být odlišný pro každý zvuk, tím se zvyšuje náročnost a výpočetní čas výpočtu.

$$r_m = \sum_{k=1}^p \varphi_k r_{m-k} + \sigma_\epsilon^2 \delta_{m,0}, \quad (1.2)$$

kde $m = 0, \dots, p$ pro $p + 1$ rovnic, p je řád, r_m je autokovarianční funkce z X_t , ω_ϵ je standartní odchylka vstupního náhodného šumu a $\delta_{m,0}$ je funkce Kronecker delta.

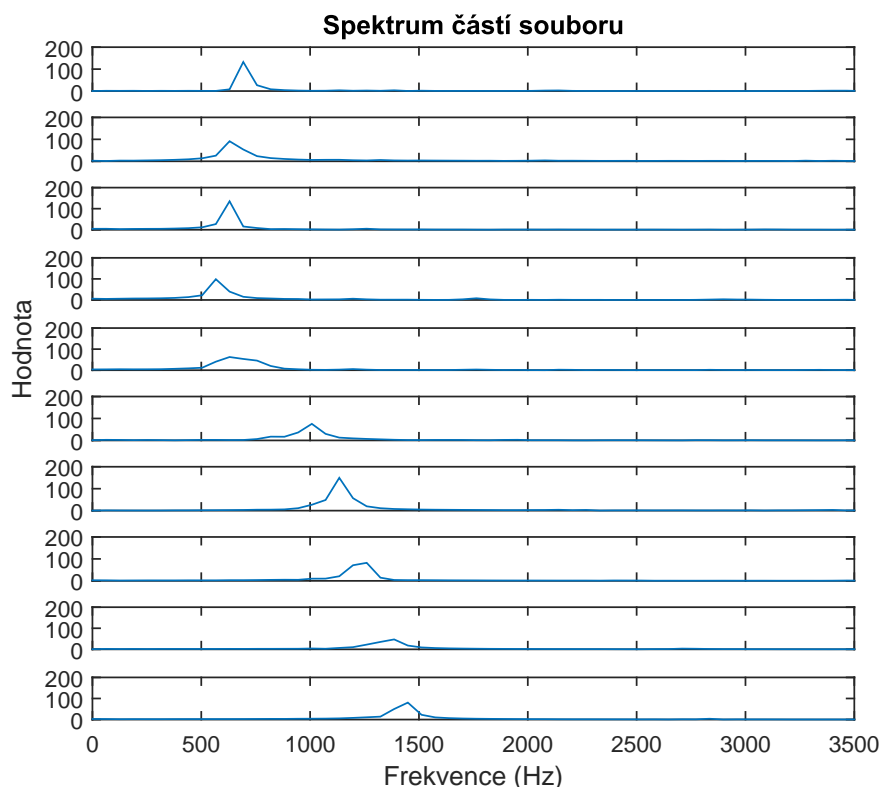
Pro tuto práci však byla vybrána FFT, která je pro svou dobrou implementovanost a výpočetní nenáročnost s výhodou využitelná na různých platformách, např. PC, mikropočítače atp. Výpočet hodnot spektra pomocí FFT je popsán vzorcem (1.3)[5].

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}, \quad (1.3)$$

kde X_k je velikost spektrální čáry, x_n je velikost vzorku v časové oblasti, N je celkový počet vzorků, n je pořadí vzorku a k je pořadí spektrální čáry ($k = 0, \dots, N - 1$).

Spektra vybraných po sobě jdoucích částí zvukového záznamu čistého zvuku sirény jsou zobrazeny na obr. 1.1, můžeme zde vidět pohyb přeladované sirény v kmitočtovém spektru.

Z kmitočtových spekter všech částí zvukového souboru jsou uchovány nejvyšší (dominantní) hodnoty, které jsou předány pro další zpracování.



Obr. 1.1: Spektra částí zvukového souboru.

1.2 Operační křivky (ROC)

Operační křivky (ROC z angl. Receiver Operating Characteristic) byly prvně využity americkou armádou během 2. světové války, při analýze radarových signálů pro rozlišení vlastních a nepřátelských vzdušných sil [7, 8]. Posléze našly ROC křivky uplatnění v medicíně [9] i veterinární [10], ale také elektrotechnice [11, 12, 13], či oblasti životního prostředí [14].

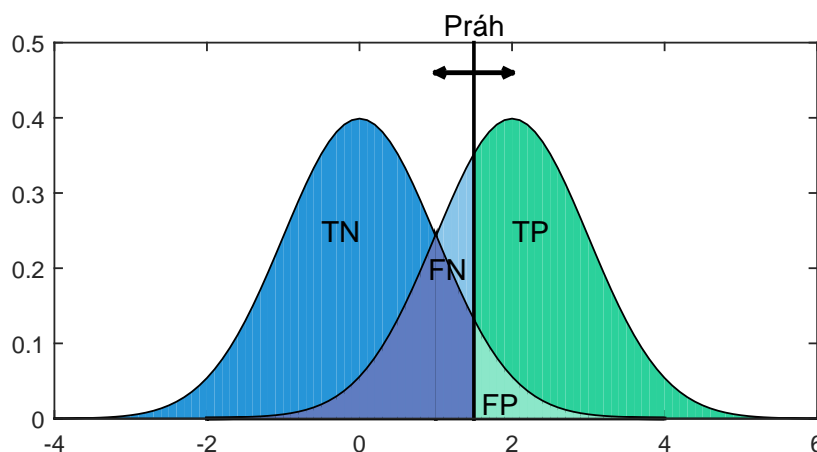
Například Bradley[11] využil ROC křivky pro vyhodnocení schopnosti samoučícího se algoritmu, kdy testoval šest algoritmů při učení šesti lékařských testů. Pro srovnání schopnosti učení pak počítal z ROC křivek AUC hodnoty. Maršálek[12] zobrazil pomocí ROC křivek schopnost detekce signálů (GSM, WiMAX, DVB-T a

MQAM) v závislosti na odstupu signálu od šumu (SNR), případně srovnání signálů při stejném SNR. Malach ve svém článku[13] využil ROC křivky pro vyhodnocení schopnosti detekce obličejů ve snímku. Le Meur využívá ROC křivky při detekci kazů (defektoskopii), kdy pomocí nich rozhoduje zda byla detekce správná či nikoli.

Výsledky testu skupiny prvků na určitou vlastnost (zařízení na funkčnost, emaily na spam, pacienti na nemoc atp.) lze zpravidla rozdělit do dvou tříd (pozitivní x negativní). Při tomto testu však mohou nastat chybná určení a tak podle znalosti stavu jednotlivých prvků je pak můžeme rozdělit do čtyř skupin [16], jak je zobrazeno na obr. 1.2. Jedná se o binormální rozložení pravděpodobností, která mají určitý překryv, obr. 1.3. Tímto překryvem je dán tvar ROC křivky, podrobný popis následuje u obr. 1.4. Určíme-li rozhodovací práh jako na obr. 1.3 dostaneme ze dvou normálních rozložení vyjádření hodnot skutečně pozitivních (TP z angl. True Positive), skutečně negativních (TN z angl. true negative), falešně pozitivních (FP z angl. false positive) a falešně negativních (FN z angl. false negative). Polohu bodů ROC křivky pak vypočítáme pomocí vztahů (1.4) a (1.5).

		Skutečná hodnota	
		pozitivní	negativní
detekovaná hodnota	pozitivní	Skutečně pozitivní TP	Falešně pozitivní FP
	negativní	Falešně negativní FN	Skutečně negativní TN

Obr. 1.2: Tabulka srovnání měřených a skutečných hodnot.



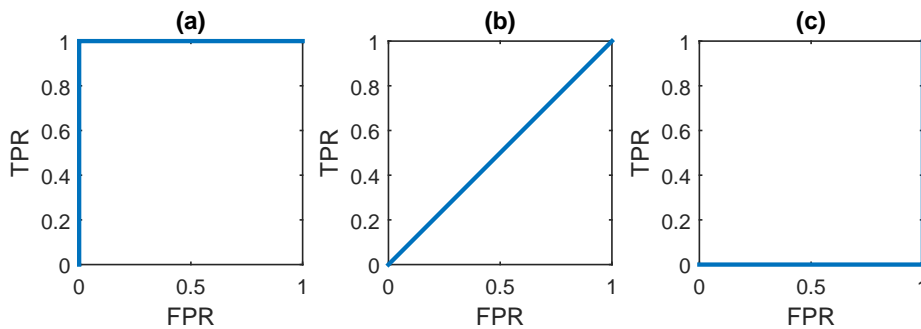
Obr. 1.3: Příklad binormálního rozdělení s prahováním.

Při analýze tedy budeme dostávat pro každý práh čtveřici hodnot, ze kterých následně pomocí vzorců 1.4 a 1.5 vypočítáme hodnoty potřebné pro vykreslení ROC křivky. Těmi jsou míra skutečně pozitivních (TPR z angl. True Positive Rate) detekcí a míra falešně pozitivních (FPR z angl. False Positive Rate) detekcí.

$$TPR = \frac{TP}{(TP + FN)} \quad (1.4)$$

$$FPR = \frac{FP}{(FP + TN)} \quad (1.5)$$

Z těchto hodnot pak vykreslíme ROC křivku, která nám ukazuje jak se mění detekce signálu při různém nastavení práhové hodnoty (případně jiného vhodného parametru). Na obr.1.4 jsou zobrazeny krajní možnosti zobrazení ROC křivek. Obrázek 1.4.a ukazuje stav, kdy byl signál detekován ve všech případech (překryv obou rozložení pravděpodobností je minimální), obrázek 1.4.b ukazuje nulovou detekci, tedy stav, kdy se oba vstupní signály sobě rovnají (rozdělení pravděpodobností se překrývají) a obrázek 1.4.c dokládá stav, kdy byl signál detekován pouze v signálu, kde detekován být neměl. Dostaneme-li tedy ROC křivku ve spodním trojúhelníku, můžeme otočit testovací kritérium, čímž křivku překlopíme dle diagonály do horní poloviny prostoru. Vyhodnocení ROC křivek se provádí pomocí plochy pod nimi (AUC - Area Under Curve – plocha pod křivkou).



Obr. 1.4: Limitní případy ROC křivek - (a) vše detekováno správně; (b) 50% úspěšnost detekce; (c) vše detekováno špatně.

Na obr.A.1 v příloze A si můžeme ukázat jak je ROC křivka konstruována. Budeme-li posouvat prahem, jak je znázorněno v grafech prvního sloupce, můžeme vypočítat integrály křivek rozdělení od prahu výše, čímž dostaneme hodnotu potřebnou k vykreslení ROC křivky. Tyto hodnoty pak vynášíme do čtvercového prostoru s osami od 0 do 1. Prahem tedy posouváme od nejnižší hodnoty po hodnotu nejvyšší, abychom obsáhli všechny prvky obou množin pravděpodobností.

1.3 Plocha pod křivkou (AUC)

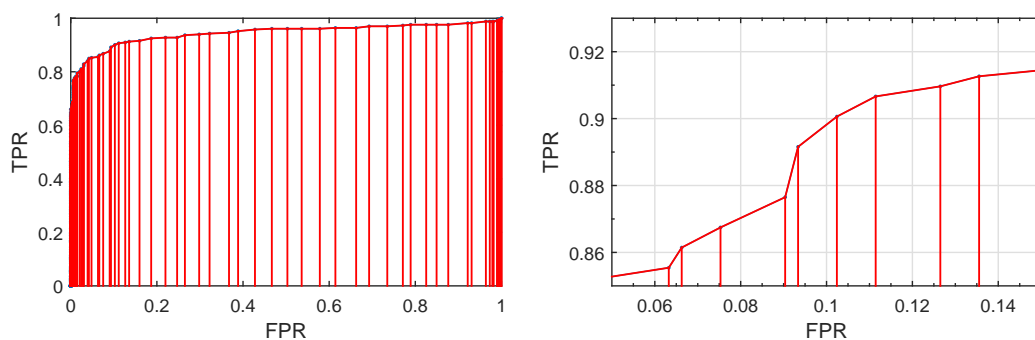
Pro porovnání ROC křivek se s úspěchem používá plocha pod ROC křivkou, označovaná jako AUC. Možností výpočtu hodnoty plochy pod křivkou je více, lze využít lichoběžníkovou metodu, obdélníkovou metodu, výpočet integrálu křivky, případně regresní křivky a mnoho dalších[11, 17, 18]. Bradley ve svém článku o hodnocení strojového učení[11] používá pro výpočet AUC lichoběžníkové pravidlo numerické integrace, jak je psáno v jeho článku. Kdy ve vzdálenosti dvou bodů (na ose x) odečítá jejich hodnoty a pracuje s jejich průměrem. Podobně Yeh ve svém příspěvku o výpočtu AUC[17] využívá lichoběžníkového pravidla pro výpočet AUC na datech převzatých pro časový vývoj koncentrace plasmy. Nakonec i Kutálek se ve své práci[19] zmiňuje především o lichoběžníkové metodě a pak také o ploše pod binormální křivkou, přičemž zde se jedná o výpočet AUC přímo z binormálního rozložení pravděpodobností bez nutnosti výpočtu ROC křivek.

Nejjednodušší metodou je výpočet integrálu, problémem však je nespojitý charakter ROC křivky. Integrální výpočet hodnoty AUC by se dal provést pomocí využití regresní křivky, zde však narážíme na nutnost využití regrese vysokého řádu pro ROC křivku s vyšším počtem bodů, abychom tuto ROC křivku přesně aproximovali a tím minimalizovali nepřesnost výpočtu. Tento výpočet je však náročnější a tedy špatně aplikovatelný do jednoduchých a levných zařízení typu mikroprocesorů apod.

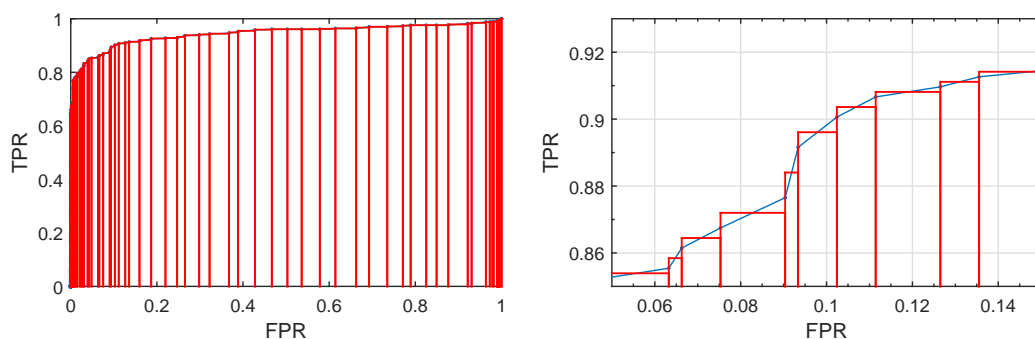
Jelikož naše ROC křivka je nespojitá a tedy složená z jednotlivých známých bodů, bude toho využito pro výpočet hodnoty AUC. Známe-li body křivky a potřebujeme zjistit plochu pod ní, je možné využít lichoběžníkovou metodu, kdy vypočítáme plochu lichoběžníku mezi sousedními body na ose x [21]. Princip lichoběžníkové metody je zobrazen na obr. 1.5. Tato metoda se však dá zjednodušit na výpočet prostého obdélníku, jehož šířka odpovídá vzdálenosti a výška střední hodnotě dvou bodů. Princip je zobrazen na obr. 1.6. Výsledek lichoběžníkové a obdélníkové metody je stejný. Zmíněná obdélníková metoda bude dále využita pro výpočet AUC, tudíž jí bude věnován větší prostor.

Pro výpočet AUC pomocí lichoběžníkové metody je potřeba znát jednotlivé body průběhu křivky. Pokud je neznáme, lze křivku rozdělit na např. ekvidistanční úseky, ve kterých nalezneme hodnoty ROC křivky a z nich vypočítáme plochu pod křivkou. Jednou z možností výpočtu je úprava hodnoty v intervalu na obdélník, čím se plocha pod křivkou v intervalu nezmění, jelikož využijeme střední hodnoty mezi krajními hodnotami intervalu. Příklad výpočtu AUC obdélníkovou metodou je zobrazen na obr.1.6. AUC je pak počítána dle vzorce (1.6).

$$AUC = \sum_{n=1}^{N-1} \left(\frac{y_n + y_{n+1}}{2} \cdot (x_{n+1} - x_n) \right), \quad (1.6)$$



Obr. 1.5: Výpočet AUC lichoběžníkovou metodou - vlevo: pohled na celou ROC křivku; vpravo: detail výpočetního intervalu lichoběžníkové metody.



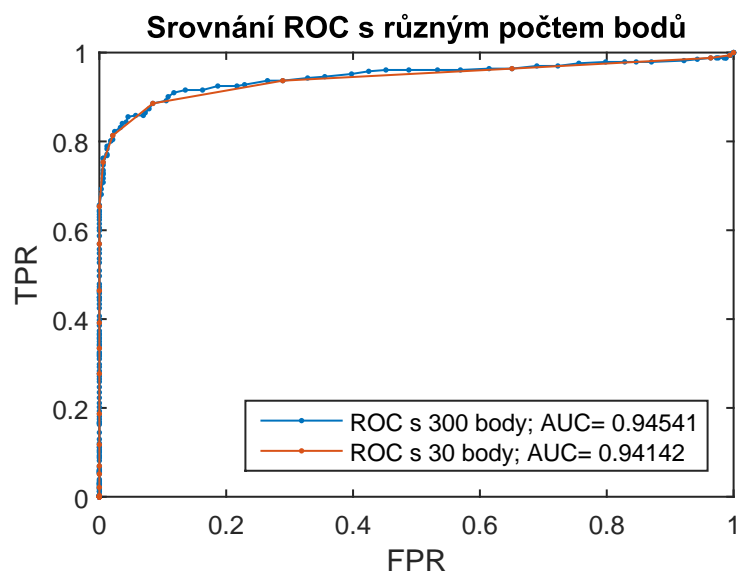
Obr. 1.6: Výpočet AUC obdélníkovou metodou - vlevo: pohled na celou ROC křivku; vpravo: detail úpravy výpočetního intervalu na obdélník.

kde $n = 1, \dots, N - 1$, N je celkový počet bodů ROC křivky, y_n je hodnota bodu na ose y a x_n je hodnota bodu na ose x .

Lze tedy říci, že čím více bodů pro sestavení ROC křivky vypočítáme a využijeme, tím bude křivka a následně i hodnota AUC přesnější. Křivky s nižším počtem bodů nebudou mít takovou přesnost a hodnota AUC také nebude přesná. Pro srovnání je na obr.1.7 porovnání dvou ROC křivek stejných vstupních dat s rozdílným počtem bodů. Pro křivky s malým počtem bodů bude výsledek této metody velmi orientační a nepřesný. Pro tyto křivky lze využít výpočet pomocí regresní křivky, jejíž průběh však bude od vypočítané ROC křivky odlišný. Takto získané hodnoty AUC budou odpovídat tvaru regresní křivky místo původní ROC křivky, takže mohou být v závislosti na odlišnosti obou křivek orientační. Dále můžeme konstatovat souvislost mezi tvarem ROC křivky a hodnotou AUC, tedy čím ostřejší ohyb ROC křivky, tím vyšší hodnota AUC a také lepší detekce žádoucího signálu.

Podle [22] lze také využít slovní hodnocení přesnosti testu (ROC křivky) dle hodnoty AUC, viz tab.1.1.

Z tabulky vyplývá, že je naší snahou optimalizovat algoritmus detekce signálu



Obr. 1.7: Srovnání ROC křivek s různým počtem bodů.

hodnota AUC	slovní hodnocení
<1 - 0,9)	výborně
<0,9 - 0,8)	velmi dobře
<0,8 - 0,7)	dobře
<0,7 - 0,6)	dostatečně
<0,6 - 0,5)	nedostatečně

Tab. 1.1: Tabulka slovních hodnocení testů dle AUC [22]

tak, aby výsledné hodnoty AUC byly co možná nejvyšší. Také vidíme, že hodnoty AUC menší než 0,5 nejsou hodnoceny. Jejich hodnocení lze ale provést po otočení rozhodovacího kritéria, poté bude AUC větší než 0,5.

2 VSTUPNÍ DATA

Pro analýzu bylo z internetové databáze zvuků Soundbible.com[23] vybráno pět zvuků sirény a několik dopravních prostředí, které byly smíchány v různých poměrech, aby bylo dosaženo vhodné simulace různých dopravních situací (běžný provoz, dopravní zácpa atp.), k těmto zvukům byl dále přidán bílý šum o úrovni -20 dB. Jejich seznam je v tabulce 2.1. Další soubor zvuků byl vytvořen smícháním zvuku sirény pouze s bílým šumem různých úrovní, pro analýzu chování algoritmu při různých úrovních odstupů signálu od šumu (SNR), jejich seznam je uveden v tabulce 2.2.

Tab. 2.1: Soubory s dopravním ruchem

soubor	popis
s.wav	pouze siréna
sw.wav	siréna s bílým šumem
st.wav	siréna s dopravním ruchem
stw.wav	siréna s dopravním ruchem a bílým šumem
sh.wav	siréna se silným dopravním ruchem
shw.wav	siréna se silným dopravním ruchem a bílým šumem
sv.wav	siréna s velmi silným dopravním ruchem
svw.wav	siréna s velmi s. dopravním ruchem a bílým šumem a stejné zvuky bez sirény

Tab. 2.2: Soubory jen s bílým šumem

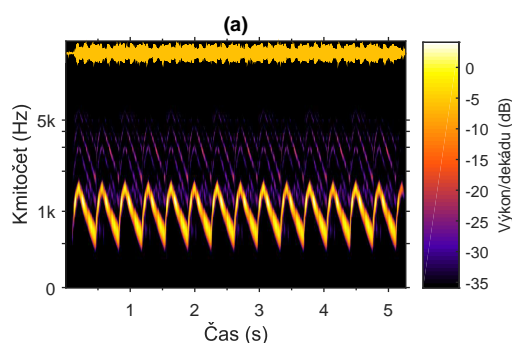
soubor	popis
s.wav	pouze siréna
sw1.wav	siréna s bílým šumem o úrovni -20 dBFS
sw2.wav	siréna s bílým šumem o úrovni -10 dBFS
sw3.wav	siréna s bílým šumem o úrovni -8 dBFS
sw4.wav	siréna s bílým šumem o úrovni -6 dBFS
sw5.wav	siréna s bílým šumem o úrovni -3 dBFS
sw6.wav	siréna s bílým šumem o úrovni -1 dBFS a stejné zvuky bez sirény

Pro vytvoření odpovídající ROC křivky je potřeba dvou zvuků podobných parametrů (zejména vzorkovací kmitočet, bitová hloubka, kmitočtový rozsah a délka), kdy jedním je zvukový soubor obsahující sirénu s rušným pozadím a druhým je soubor obsahující rušné pozadí samotné. V reálném světě však nastává problém získání

dvou zvuků s naprosto stejným zvukovým pozadím. Toto se dá realizovat využitím dvou krátkých zvukových záznamů po sobě následujících v čase, kdy jeden je získán v čase kdy siréna ještě nezněla a druhý v čase kdy již byla slyšet siréna (např. průjezd vozidla IZS), musí si však být tyto zvuky časově velmi blízké, aby bylo zaručena podobnost zvukového pozadí.

Použité zvukové soubory jsou ve formátu WAVE s vzorkovacím kmitočtem 44,1 kHz, bitovou hloubkou 16 bitů a datovým tokem 706 kbps.

Spektrogramy zvukových souborů jsou umístěny v přílohách B až H. Pro názornost si jeden popíšeme, je zobrazen na obr.2.1. V horní části je zobrazena amplitudová obálka zvuku v průběhu času. Pod ní je zobrazen spektrogram, kdy na svislé ose je vynesena kmitočet v Hertzích a na vodorovné je čas v sekundách. Amplituda daného kmitočtu je zobrazena barevně, podle stupnice na pravé straně obrázku.



Obr. 2.1: Zobrazení amplitudové obálky a spektrogramu zvukového souboru.

3 APLIKACE

3.1 Konzolová aplikace

Principy analýzy jsou popsány v kapitole 1, nyní se bude práce zabírat praktickou stránkou, tedy výpočetním skriptem ve vývojovém prostředí Matlab. Pro tvorbu skriptu byly využity implementované funkce Matlabu a také toolbox VoiceBox[24]. Výpočetní skript se skládá ze tří `m-file` souborů prostředí Matlab. Prvním je funkce `analyza`, která provádí načtení zvukového souboru, jeho rozdělení na dílčí vektory, FFT analýzu a výběr nejvyšších hodnot spekter. Ve druhém souboru je funkce `roc`, která provádí výpočet bodů ROC křivky. A třetím souborem jsou volány funkce `roc` pro výpočet ROC křivek, dále jejich zobrazení a výpočet AUC ploch. Všechny soubory jsou přiloženy v přílohách I, J a K

```
function [FFTmax]=analyza(file,lsample)
%% nahrani zvukoveho souboru, jeho rozkouskovani a FFT
% vyuziva toolbox Voicebox

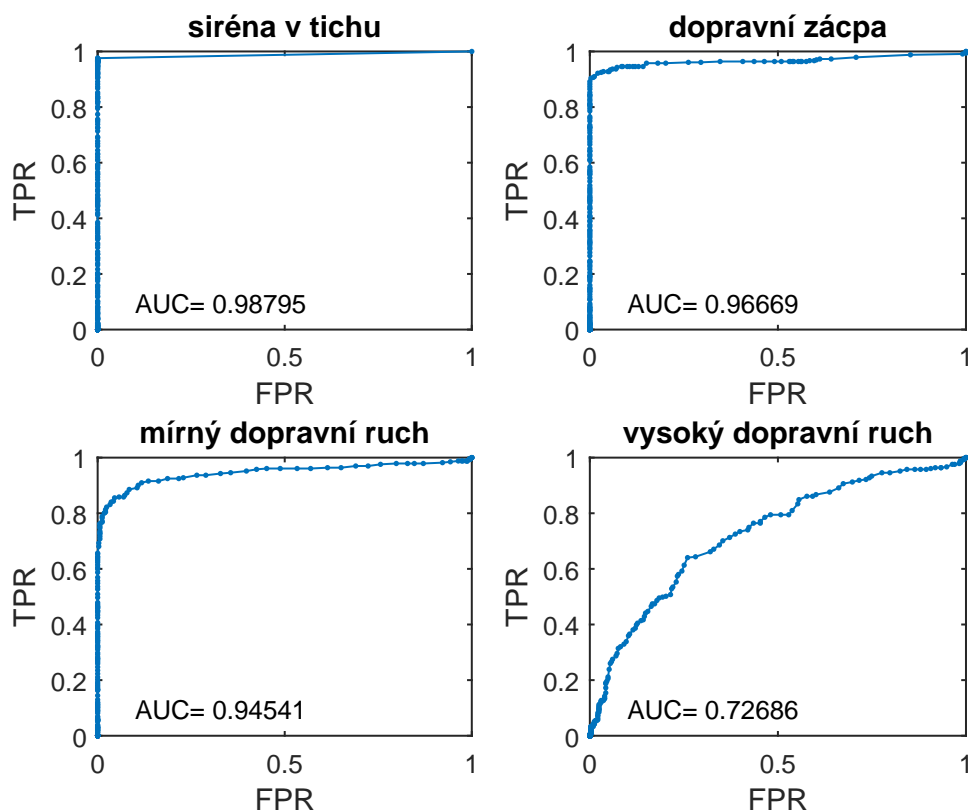
[data,fs]=readwav(file);           % nacteni celeho zvukoveho souboru

%% rozdeleni zvukoveho souboru na casti o delce 'lsample'
for d = 1:(length(data)/lsample)   % vypocet poctu dilu souboru
    [data2]=readwav(file,'',lsample,(d-1)*lsample);
    % nacteni urcitych vzorku ze souboru
    FFTmax(d)=max(abs(rfft(data2)));
    % FFT analiza a ulozeni nejvyssi hodnoty
end
```

Prvním krokem výpočetního skriptu je načtení souboru, pro které je využita funkce `readwav` toolboxu `Voicebox`. Tato funkce je určena k načítání zvukových souborů ve formátu `Wav` do prostředí Matlab. Kromě načtení zvuku a jeho uložení do určeného vektoru zjistí i další parametry souboru jako je bitová hloubka, vzorkovací kmitočet nebo počet kanálů, také zobrazí časovou obálku zvuku a jeho spektrogram, který je zobrazen např. na obr.2.1.

Načtený zvukový soubor je dále rozdělen na úseky délky 700 vzorků, na kterých byla provedena Rychlá Fourierova transformace (FFT). Pro FFT je využita funkce implementována v prostředí Matlab, jak je popsáno v kap. 1.1 na straně 3. Délka 700 vzorků byla zvolena po experimentálním testování, při kterém byl testován tvar spektra stejného úseku zvuku na délce tohoto úseku. Při délce 700 vzorků byl dosažen kompromis mezi filtrací rušení a úrovní užitečného signálu (hledané sirény). Při délce <700 bylo kmitočtové spektrum složeno s malého počtu bodů a bylo tedy nepřesné,

při hodnotě >700 bylo spektrum detailnější, avšak úsek zvukového záznamu byl delší a tudíž přeladovaná siréna zabírala širší část spektra. Spektra několika částí jsou zobrazena na obr. 1.1. Při testování omezení šířky pásma zvukového souboru, příp. omezení vypočítaného kmitočtového spektra signálu, nedošlo k žádným změnám v ROC křivkách a hodnotách jejich AUC.



Obr. 3.1: Vypočítané ROC křivky

```
function [Pd,Pfa]=roc(data1,data2)
%funkce pro vypocet a vykresleni ROC krivky
%autor: Jitka Pomenkova
%data1=lsig_1;...signal se sumem
%data2=lsum_1;...sum

%=====cyklus pro vypocet hodnot Pd a ...
    Pfa=====
K1=sum(data1)/length(data1);
krok=K1/100;
hodnoty=0:krok:K1*3;
for j=1:length(hodnoty)
    K=hodnoty(j);
```

```

P(j,1)=K;

%-----SIGNAL+SUM:sumaksig2
%pocitam dobrou detekci
%data1
T=0;F=0;
    for i=1:length(data1)
        if data1(i)>K % nasla jsem signal
            B(1,i)=1; T=T+1;
        else B(1,i)=0; F=F+1;
        end
    end
    Pd(1,j)=T/(T+F);

%-----SUM:sumak2
%pocitam falesnou detekci
%data2
Tt=0;Ft=0;
    for i=1:length(data2)
        if data2(i)>K %netrefila jsem se (nasla jsem signal), tj. ...
            opravdu falesna detekce
            A(1,i)=1; Tt=Tt+1;
        else A(1,i)=0; Ft=Ft+1;
        end
    end
    Pfa(1,j)=Tt/(Tt+Ft); %tj. pst, kdy jsem se netrefila v ...
        identifikaci sumu

end
plot(Pfa,Pd, '-','Marker','.')

```

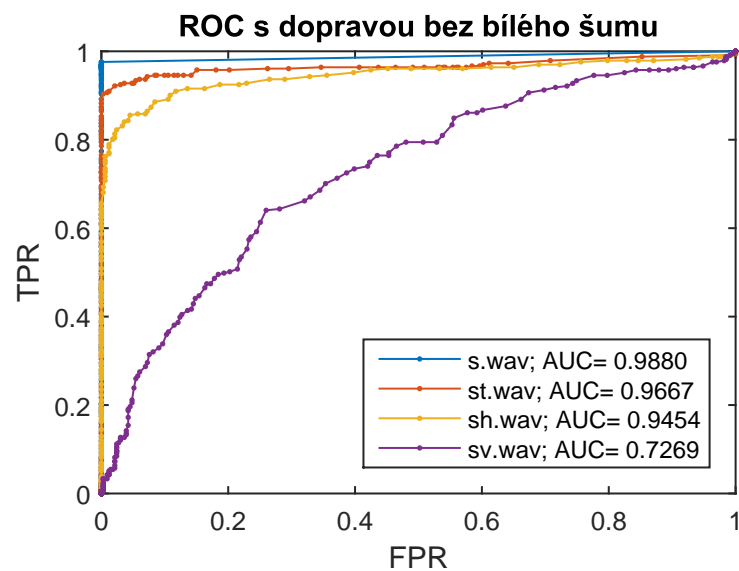
Z každého získaného spektra je zachována nejvyšší hodnota, která je využita pro výpočet ROC křivky. Pro výpočet ROC křivky je potřeba dvou zvuků, jeden obsahující zvuk sirény s rušným pozadím a druhý obsahující pouze zvuk pozadí. Z vektorů maximálních hodnot spekter dvojice zvuků jsou posléze pomocí posuvného prahu získány hodnoty TP, TN, FP a FN, ze kterých jsou vypočítány hodnoty TPR a FPR (viz kap. 1.2). ROC křivky jsou složeny celkem z 300 bodů. Hodnoty TPR a FPR jsou následně vykresleny do grafu jako ROC křivka. Princip získání ROC křivky je podrobně popsán v kapitole 1.2 na straně 3. Z tvaru křivky lze určit odlišnost dvou vstupních zvuků; blíží-li se křivka levému hornímu rohu (bodu [0;1]) jsou oba zvuky velmi rozdílné, pokud se křivka blíží diagonále jsou si zvuky velmi podobné. Na obr.3.1 jsou zobrazeny ROC křivky pro 4 zvuky se stejnou sirénou, ale odlišnými úrovněmi pozadí. Pro lepší možnost srovnání je vhodné zobrazit dané křivky v jednom grafu, jak je zobrazeno na obr. 3.2.

```

%% vypocet AUC
[Pd,Pfa]=roc(respol,resnic);    % vypocet ROC krivky a hodnot Pd a Pfa
AUC(1)=0;
for a = 1:length(Pfa)-1        % vypocet poctu prubehu cyklu
    AUC(1)=AUC(1)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
                                % vypocet AUC plochy dle vzorce
end

```

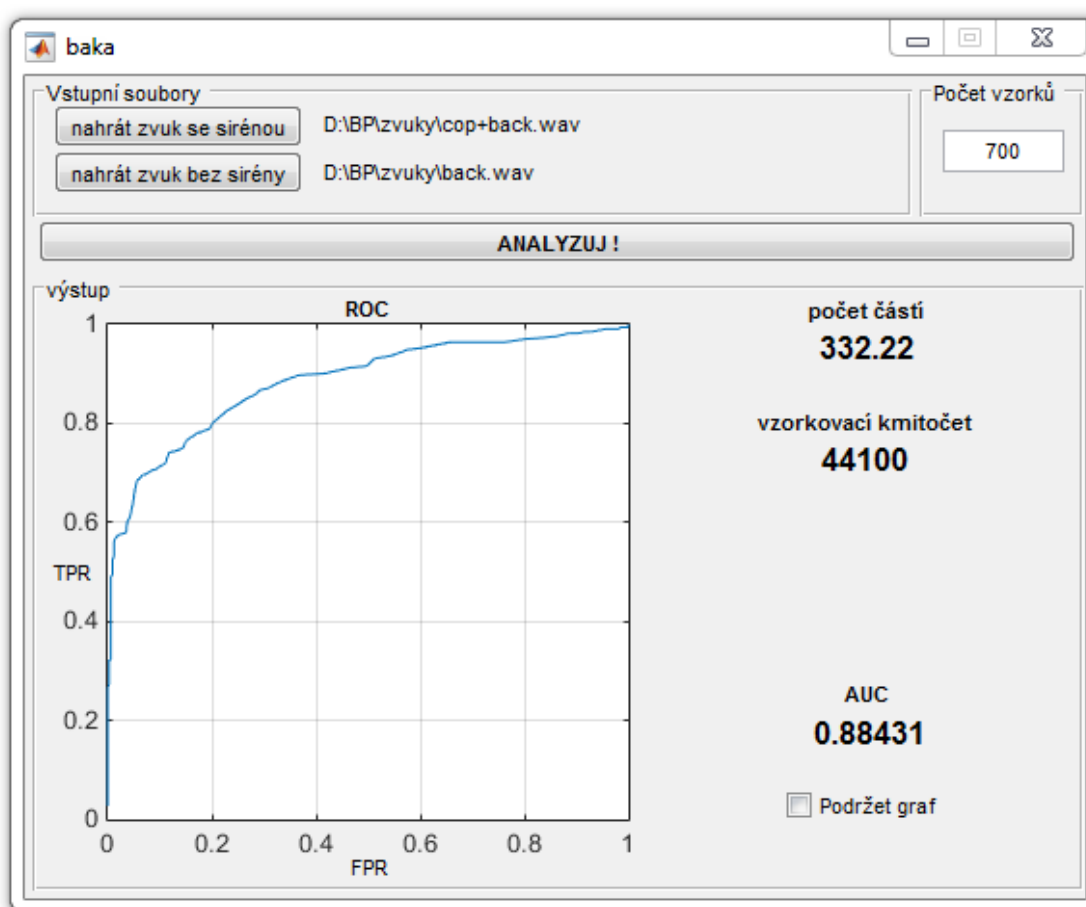
Pro lepší přehled a možnost srovnání několik ROC křivek je vypočítána hodnota plochy pod každou křivkou (AUC). Výpočet hodnoty AUC je proveden pomocí obdélníkové metody, která byla popsána v kapitole 1.3 na str. 8. Taktéž všechny hodnoty AUC v obrázcích, resp. tabulkách, v této práci jsou vypočteny pomocí obdélníkové metody.



Obr. 3.2: ROC křivky zobrazené v jednom grafu, pro lepší srovnání

3.2 Grafické rozhraní

Z algoritmu z předchozí kapitoly 3.1 byla vytvořena aplikace s jednoduchým uživatelským rozhráním. Aplikace umožňuje načtení dvojice zvuků ve formátu WAV. Dále umožňuje nastavení počtu vzorků jednoho dílu, na který jsou vstupní zvukové soubor rozděleny. Stiskem tlačítka „ANALYZUJ!“ provede program výpočet a zobrazení ROC křivky a hodnoty její AUC plochy, vypočítané pomocí obdélníkové metody. Posledním prvkem grafického rozhraní je zaškrťovací políčko „Podržet graf“, které umožňuje podržet na obrazovce současný graf, vybrat další dvojici souborů a vykreslit je do stejného obrázku.



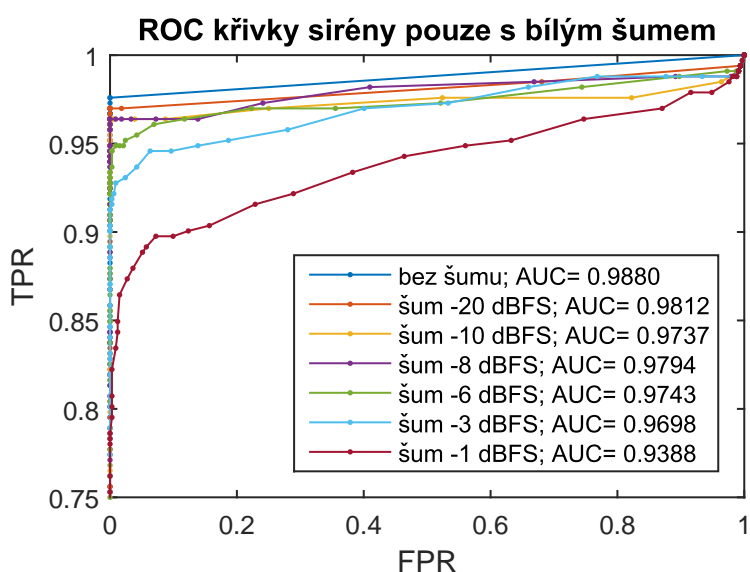
Obr. 3.3: Grafické rozhraní programu

4 VÝSLEDKY

Analýza byla provedena na pěti zvucích sirény v kombinaci s rušnými pozadími. Hodnoty AUC zobrazené v grafech a tabulkách jsou vypočítány pomocí obdélníkového pravidla popsaného na straně 8 v kap. 1.3. Jednotlivé výsledky budou popsány v následujících odstavcích. Výsledné srovnání a tabulka všech hodnot je v poslední podkapitole 4.3.

4.1 Siréna s bílým šumem

Pro první analýzu byla vybrána první siréna a byl zkoumán vliv úrovně bílého šumu na schopnost detekce sirény ve zvuku. Nejvyšší úroveň šumu byla zvolena -1 dBFS, při které ještě nedocházelo k digitálnímu přebuzení audio signálu. Spektrogramy jsou zobrazeny na obr.B.1 v příloze B. Z těchto signálů byly vypočítány ROC křivky a následně hodnoty AUC, ty jsou zobrazeny na obr. 4.1 (pro větší přehlednost, bylo změněno měřítko svislé osy. Z hodnot AUC a průběhů ROC křivek lze říci, že i vysoká hodnota zašumění (resp. nízká hodnota SNR) nemají na detekci zvuku sirény velký vliv, změna ROC je menší než jedna desetina.



Obr. 4.1: ROC křivky zvuků sirény s bílým šumem různých úrovní

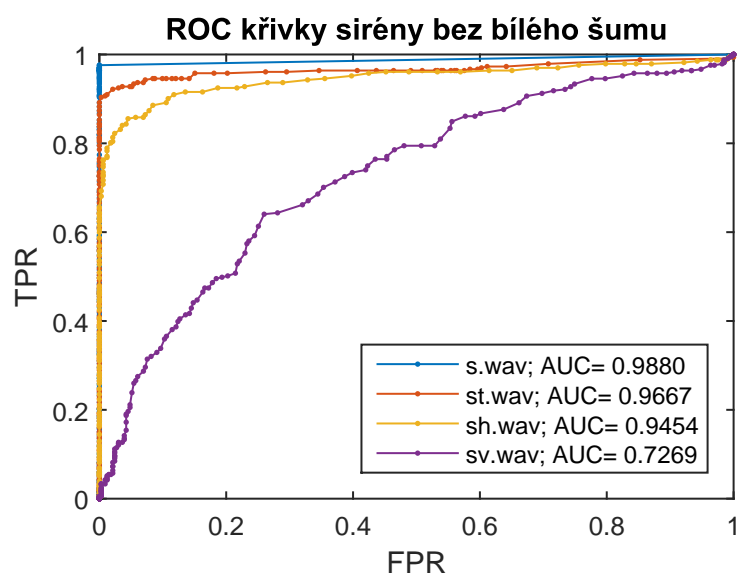
4.2 Sirény s dopravním ruchem

Jako hlavní část byla analýza několika zvuků sirén v různých dopravních prostředcích. Jednotlivé použité zvuky budou vždy zobrazeny nejen pro představu „utopení“

sirény v ruchu pozadí. Samotné analyzované zvuky jsou popsány v následujících podkapitolách.

4.2.1 První siréna

První sirénou byla policejní siréna. Spektrogramy zvuků sirény s dopravním rušením, resp. s přidaným bílým šumem o úrovni -20 dBFs jsou zobrazeny na obr. C.1 a D.1 v přílohách C a D. Z těchto zvuků byly vypočítány ROC křivky a následně hodnoty AUC, tyto jsou zobrazeny na obr. 4.2, resp. obr. 4.3.



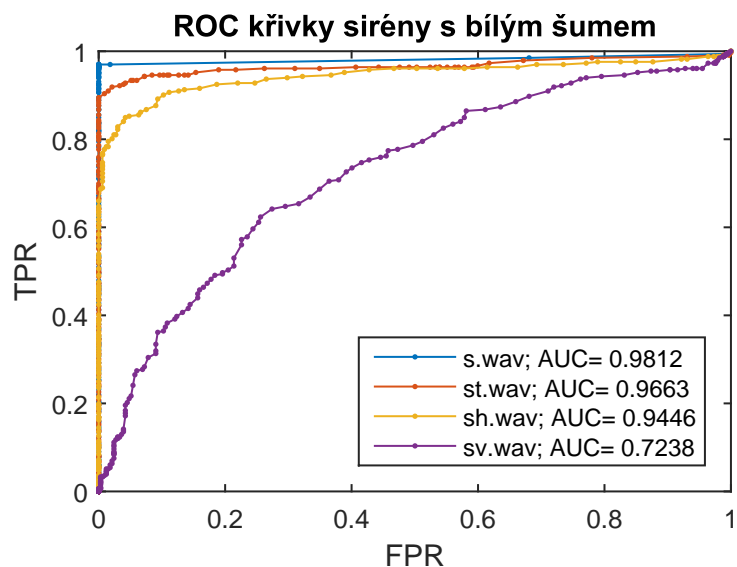
Obr. 4.2: ROC křivky zvuků první sirény s rušným pozadím

Srovnáním obrázků 4.2 a 4.3 lze říci, že přidání bílého šumu úrovně -20 dBFs provede pouze minimální změnu ve tvaru křivky, hodnotě AUC a také v účinnosti detekce sirény, jak dokazuje srovnání v tabulce 4.1. Proto nebudou u dalších zvuků sirén zobrazovány ROC křivky pro zvuky s dopravním ruchem a bílým šumem.

zvuk	AUC bez šumu	AUC se šumem	dAUC
s.wav	0,9880	0,9812	0,0068
st.wav	0,9667	0,9663	0,0004
sh.wav	0,9454	0,9446	0,0008
sv.wav	0,7269	0,7238	0,0031

Tab. 4.1: Tabulka srovnání AUC křivek bez bílého šumu a s ním dle obr. 4.2 a 4.3

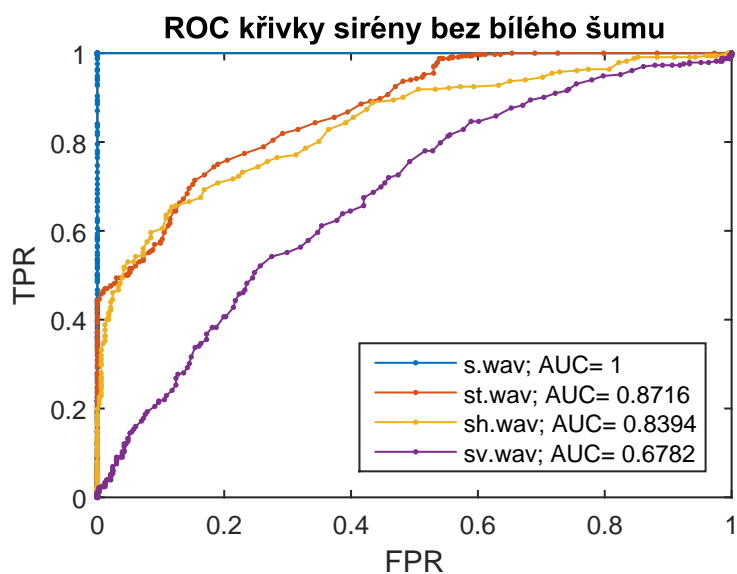
Zkoumáním ROC křivek můžeme říci, že lze sirénu detekovat i ve vysoce zarušeném prostředí, kde je nejvíce rušení v oblasti kmitočtů sirény. Hodnota AUC zde dosahuje 0,7238, což lze dle tabulky 1.1 hodnotit jako dobrý výsledek.



Obr. 4.3: ROC křivky zvuků první sirény s rušným pozadím a bílým šumem

4.2.2 Druhá siréna

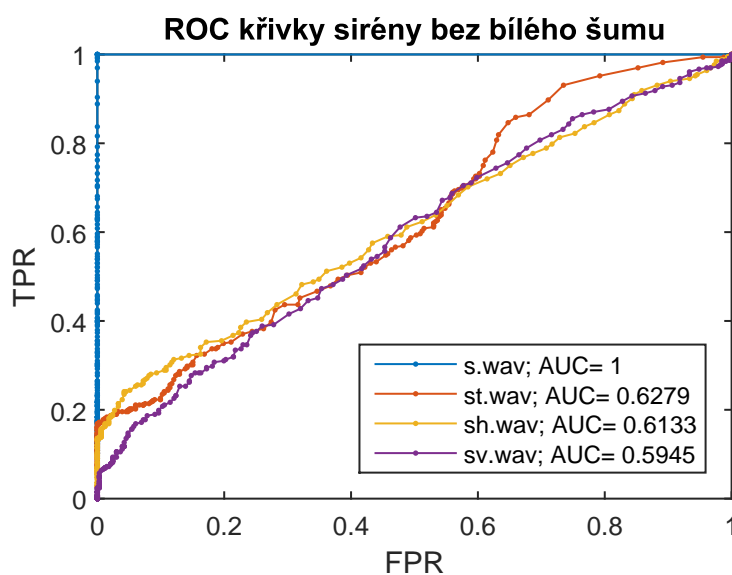
Záznam druhé sirény je z reálného provozu a obsahuje tedy na pozadí přirozený (nepřidaný) dopravní ruch, spektrogramy jsou na obr. E.1 v příloze E. Přidáním dalších ruchů dopravního pozadí, roste úroveň rušivého pozadí a tedy klesá hodnota AUC. Pro ruch dopravní zácpy a mírného dopravního prostředí se pohybuje v okolí 0,85, při velmi rušném dopravním prostředí klesne k 0,68, což lze považovat za dobrý výsledek detekce. ROC křivky s hodnotami AUC jsou zobrazeny na obr. 4.4.



Obr. 4.4: ROC křivky zvuků druhé sirény s rušným pozadím

4.2.3 Třetí siréna

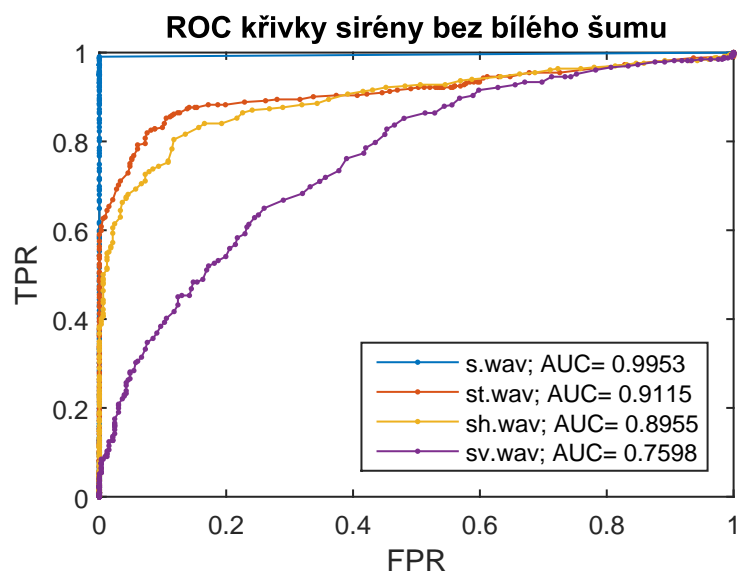
Zvuk třetí sirény je více rozprostřen ve kmitočtovém spektru, proto je lépe maskován dopravním ruchem, spektrogramy jsou zobrazeny na obr. F.1 v příloze F. Přidáním dopravních ruchů tedy dojde k velkému poklesu hodnoty AUC a také účinnosti detekce. Pro přidaná pozadí dopravní zácpy, mírného i velmi rušného dopravního prostředí se pak hodnoty AUC pohybují okolo 0,6, což odpovídá „utopení“ zvuku sirény v okolním prostředí. Výsledek se tedy pohybuje na rozmezí dostatečné a nedostatečné klasifikace, výsledek tedy nelze považovat za průkazný. ROC křivky jsou zobrazeny na obr. 4.5.



Obr. 4.5: ROC křivky zvuků třetí sirény s rušným pozadím

4.2.4 Čtvrtá siréna

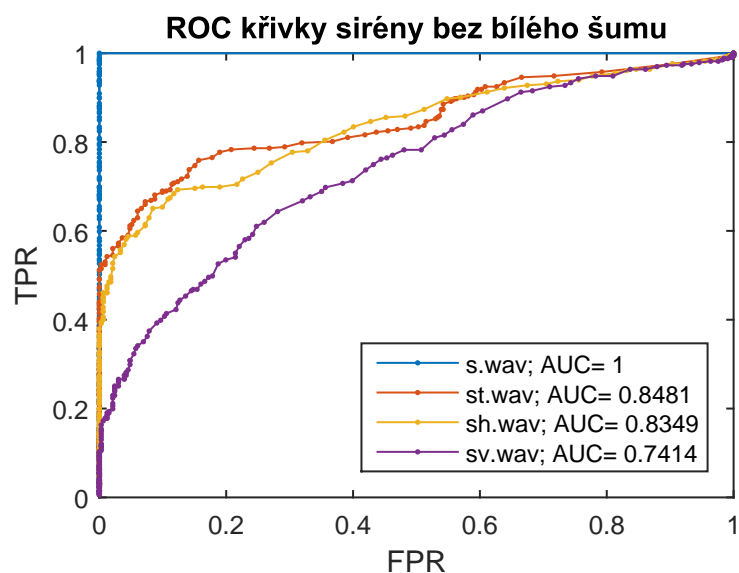
Čtvrtá siréna obsahuje kromě základního tónu také jeho druhý harmonický kmitočet, zabírá tedy také větší část akustického pásma, spektrogramy jsou zobrazeny na obr. G.1 v příloze F. Proti předchozí (třetí) siréně je však její energie soustředěna do užšího spektra v jednom okamžiku, proto je lépe detekovatelná. Hodnoty AUC pro prostředí dopravní zácpy a mírné dopravy se pohybují kolem 0,9, při silné dopravě klesají k 0,75. Siréna je tedy v simulovaných prostředích dobře detekovatelná. ROC křivky s hodnotami AUC jsou zobrazeny na obr. 4.6.



Obr. 4.6: ROC křivky zvuků čtvrté sirény s rušným pozadím

4.2.5 Pátá siréna

Pátá siréna obsahuje převážně hlavní kmitočet a některé vyšší harmonické. Spektrogramy jsou zobrazeny na obr. H.1 v příloze G. Pro rušivé pozadí dopravní zácpy a mírné dopravy je hodnota AUC blízka 0,83, při rušném dopravním prostředí poklesne AUC k hodnotě 0,75. ROC křivky jsou spolu s hodnotami AUC zobrazeny na obr. 4.7.



Obr. 4.7: ROC křivky zvuků páté sirény s rušným pozadím

4.3 Srovnání

šum	hodnota AUC					poznámka
	první siréna	druhá siréna	třetí siréna	čtvrtá siréna	pátá siréna	
bez šumu	0,9880	1,0000	1,0000	0,9953	1,0000	s.wav
dopravní ruch	0,9667	0,8716	0,6279	0,9115	0,8481	st.wav
dopravní ruch + b.š.	0,9663	0,8686	0,6271	0,9106	0,8487	stw.wav
silný ruch	0,9454	0,8394	0,6133	0,8955	0,8349	sh.wav
silný ruch + b.š.	0,9446	0,8383	0,6140	0,8931	0,8354	shw.wav
velmi silný ruch	0,7269	0,6782	0,5945	0,7598	0,7414	sv.wav
velmi silný ruch + b.š.	0,7238	0,6796	0,5947	0,7578	0,7404	svw.wav
bílý šum -20 dBFS	0,9812	1,0000	0,9719	0,9723	0,9701	sw1.wav
bílý šum -10 dBFS	0,9737	0,9939	0,7682	0,9492	0,9248	sw2.wav
bílý šum -8 dBFS	0,9794	0,9841	0,7511	0,9521	0,9238	sw3.wav
bílý šum -6 dBFS	0,9743	0,9640	0,7076	0,9504	0,8938	sw4.wav
bílý šum -3 dBFS	0,9698	0,8907	0,6603	0,9286	0,8449	sw5.wav
bílý šum -1 dBFS	0,9388	0,7631	0,5937	0,8315	0,7679	sw6.wav

Tab. 4.2: Tabulka srovnání hodnot AUC všech zvuků

Pozorováním výsledků lze říci, že uvedený algoritmus je schopen úspěšně detekovat zvuk sirény v dopravním prostředí s dobrou účinností. V tabulce 4.2 jsou hodnoty AUC všech analyzovaných zvuků. Z dvojic stejných sirén se stejným dopravním pozadím avšak bez přidaného bílého šumu, resp. s ním, lze vypočítat mírně horší detekovatelnost sirény po přidání bílého šumu, rozdíl je však minimální (v řádu 10^{-3} až 10^{-4}). Srovnáním sirén mezi sebou můžeme vidět velmi dobrý výsledek pro sirénu v tichu, dosahující hodnoty 1,0 a při zvyšující se úrovni rušivého pozadí hodnota AUC postupně klesá k hodnotám pohybujícím se na hranici užitečnosti testu (rozmezí AUC 0,7 (dostatečný) až 0,5 (nedostatečný)). Při analýze vlivu pouze bílého šumu na účinnost detekce byly změny AUC při snižujícím se odstupu signálu sirény od šumu v řádech 10^{-2} až 10^{-1} pro rozmezí úrovně bílého šumu -20 dBFS až -1 dBFS. Vždy však záleží na úrovni sirény a rozložení její energie ve spektru. Třetí siréna je, narozdíl od jiných, hůře detekovatelná, což je dáno jejím rozprostřením ve spektru a nižší pronikavostí.

5 ZÁVĚR

Cílem práce bylo hodnocení účinnosti detekce zvukových signálů pomocí ROC křivek. Pro detekci signálu byl zvolen signál sirény vozidel IZS v dopravním prostředí. K simulaci různých prostředí bylo vytvořeno několik kombinací zvuků sirény a rušných dopravních pozadí o různých úrovních. Detekce signálu probíhala pomocí skriptu vývojového prostředí Matlab.

Každý zvukový signál byl rozdělen na dílčí úseky, které byly analyzovány pomocí Rychlé Fourierovy transformace (FFT) a z nejvyšších hodnot spekter dílčích částí každého souboru byly vytvořeny ROC křivky. Pro lepší možnost srovnání schopnosti detekce sirény v jednotlivých zvukových souborech byla z každé ROC křivky vypočítána hodnota plochy pod křivkou (AUC). Metodika a aplikace jsou popsány v kapitolách 1 a 2.

Vypočítané ROC křivky a hodnoty jejich AUC ploch jsou zobrazeny v kapitole výsledky. Hodnota AUC se může pohybovat v hodnotách 1,0-0,5 přičemž čím vyšší hodnota, tím lepší detekce. Hodnoty sirény v tichu se blížily k dokonalé detekci a tedy hodnotě 1,0. Při mírném dopravním rušení se hodnota AUC pohybovala v rozmezí 0,9-0,7, což lze považovat za dobrý výsledek detekce. Při vysokém rušení dopravním hlukem hodnota AUC poklesla do rozmezí 0,7-0,6, kterou lze stále považovat za dostatečnou pro určení přítomnosti sirény ve zvuku.

Analýzou všech vytvořených souborů a srovnáním jejich ROC křivek, resp. hodnot ploch pod křivkou lze dojít k závěru, že zvolený algoritmus je schopen úspěšně detekovat sirénu v zaznamenaném zvuku i v hlučném dopravním prostředí. Přitom si však zachovává požadovanou jednoduchost pro snadnou implementaci do různých platforem.

K hodnocení účinnosti detekce lze tedy doporučit výpočet ROC křivky a následně hodnoty plochy pod touto křivkou, pomocí metody vhodné pro danou aplikaci, v případě této práce pomocí obdélníkové metody. Hodnoty AUC získané stejnou metodou a pocházející ze souvisejících vstupních dat (např. siréna s různými pozadími) lze mezi sebou srovnávat a jsou dobrým měřítkem pro hodnocení účinnosti detekce. Blíží-li se hodnota AUC číslu 1,0 lze výsledek detekce považovat za výborný. Pokud se AUC pohybuje mezi 0,9 a 0,75 považujeme výsledek za velmi dobrý. Dosáhne-li plocha pod křivkou hodnoty 0,75 až 0,6 dá se výsledek považovat za průkazný. Poklesne-li hodnota AUC pod 0,6 je výsledek neprůkazný, resp. nelze jeho výsledek považovat za příliš relevantní.

LITERATURA

- [1] MEUCCI, F., et al. *A real-time siren detector to improve safety of guide in traffic environment*. In: Signal Processing Conference, 2008 16th European. IEEE, 2008. s. 1-5. ISSN: 2219-5491
- [2] MIYAZAKI, T.; KITAZONO, Y.; SHIMAKAWA, M. *Ambulance siren detector using FFT on dsPIC*. In: The 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing 2013 (ICISIP2013). 2013. s. 266-269
- [3] RAMASUBRAMANIAN, V., et al. *Two-class verifier framework for audio indexing*. In: Acoustics, IEEE International Conference on Speech and Signal Processing (ICASSP) 2013 . IEEE, 2013. s. 838-842. ISSN: 1520-6149, doi: 10.1109/ICASSP.2013.6637766.
- [4] BENDLER E., et al. *Emergency vehicle detector*. 2008. Senior Design Project. Georgia School of electrical and computer engineering. [online] [cit. 2016-05-30] Dostupné z:<http://www.med.mcgill.ca/epidemiology/hanley/software/hanley_mcneil_radiology_82.pdf>
- [5] BENESTY J. *Estimation spectrale Méthodes non-paramétriques*. 28 s. [online] [cit. 2016-05-25]. Dostupné z: <<http://externe.emt.inrs.ca/users/benesty/lectures/lecture14.pdf>>
- [6] ESHEL G. *The Yule Walker Equations for the AR Coefficients*. 8 s. Wharton University of Pennsylvania. [online] [cit. 2016-05-28]. Dostupné z:<<http://www-stat.wharton.upenn.edu/~steele/Courses/956/ResourceDetails/YWSourceFiles/YW-Eshel.pdf>>
- [7] METZ, Ch. E. *Basic principles of ROC analysis*. In: *Seminars in nuclear medicine*. WB Saunders, 1978, s. 283-298. doi: [http://dx.doi.org/10.1016/S0001-2998\(78\)80014-2](http://dx.doi.org/10.1016/S0001-2998(78)80014-2), ISSN: 0001-2998.
- [8] WESTIN, L.K. *Receiver operating characteristic (ROC) analysis: Evaluating discriminance effects among decision support systems*. 28 s. [online] [cit. 2016-05-25]. Dostupné z: <<http://nutkin.cs.umu.se/research/reports/2001/018/part1.pdf>>
- [9] ZWEIG, M. H.; CAMPBELL, G. *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine*. *Clinical chemistry*, 1993, 39.4: s. 561-577.

- [10] GREINER, M.; PFEIFFER, D.; SMITH, R. D. *Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests*. Preventive veterinary medicine, 2000, 45.1: 23-41. ISSN: 0167-5877
- [11] BRADLEY, A. P. *The use of the area under the ROC curve in the evaluation of machine learning algorithms*. Pattern recognition, 1997, 30.7: s. 1145-1159. ISSN: 0031-3203, doi: [http://dx.doi.org/10.1016/S0031-3203\(96\)00142-2](http://dx.doi.org/10.1016/S0031-3203(96)00142-2).
- [12] MARŠÁLEK, R.; POMĚNKOVÁ J.; BALADA R.; POVALAČ K. *On the performance of various spectrum sensing detectors on wireless communication signals*. In: 2012 22nd International Conference Radioelektronika. Piscataway: IEEE, 2012, s. 1-4. ISBN 9781467306591.
- [13] MALACH, T.; POMĚNKOVÁ, J.. *Confidence assessment of face recognition results*. In: Radioelektronika, 2015 25th International Conference. IEEE, 2015, s. 176-179.
- [14] PETERSON, A. T.; PAPEŞ, M.; SOBERÓN, J. *Rethinking receiver operating characteristic analysis applications in ecological niche modeling*. Ecological modelling, 2008, 213.1: s. 63-72. ISSN: 0304-3800.
- [15] LE MEUR, Y.; VIGNOLLE, J-M.; CHANUSSOT, J. *Practical use of receiver operating characteristic analysis to assess the performances of defect detection algorithms*. In: Journal of Electronic Imaging, vol. 17, no. 3, 2008. s. 1-13.
- [16] FAWCETT T. *ROC Graphs: Notes and Practical Considerations for Researchers*. 38 s. HP Laboratories 2004. [online] [cit. 2016-05-26]. Dostupné z: <http://www.cs.ru.nl/~tomh/onderwijs/dm/dm_files/ROC101.pdf>
- [17] YEH S-T. *Using Trapezoidal Rule for the Area Under a Curve Calculation*. 4 s. SAS Institute Inc. [online] [cit. 2016-05-26]. Dostupné z: <<http://www2.sas.com/proceedings/sugi27/p229-27.pdf>>
- [18] CALDERS T.+ JAROSZEWICZ S. *Efficient AUC Optimization for Classification*. In: Knowledge Discovery in Databases: PKDD 2007. Springer Berlin Heidelberg, 2007. s. 42-53. [online] [cit. 2015-05-29]. Dostupné z:<<https://www.semanticscholar.org/paper/Efficient-AUC-Optimization-for-Classification-Calders-Jaroszewicz/845ab7466fe665e339575419514ae7deb84df2ab/pdf>>
- [19] KUTÁLEK D. *Statistická analýza ROC křivek*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 53 s. Vedoucí diplomové práce doc. RNDr. Jaroslav Michálek, CSc.

- [20] HANLEY J.A.+ MCNEIL B.J. *The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve*. In: Radiology vol.143, no.1, s. 29-36. 1982. [online] [cit. 2016-05-30] Dostupné z:<http://www.med.mcgill.ca/epidemiology/hanley/software/hanley_mcneil_radiology_82.pdf>
- [21] CAZELAIS G. *The Trapezoidal Rule*. 2 s. School of Arts and Science, Camosun College. [online] [cit 2016-05-29]. Dostupné z:<http://pages.pacificcoast.net/~cazelais/187/trapezoidal_rule.pdf>
- [22] PLUHÁČKOVÁ H. *Neparametrické odhady ROC křivek*. Brno: Masarykova univerzita, Přírodovědecká fakulta, 2013. 53 s. Vedoucí diplomové práce Mgr. Jiří Zelinka Dr.
- [23] *Soundbible.com: Free sound clips* [online]. 2006 [cit. 2016-05-24]. Dostupné z: <<http://www.soundbible.com>>
- [24] *VOICEBOX: Speech Processing Toolbox for MATLAB* [online]. 2006 [cit. 2016-05-24]. Dostupné z: <<http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

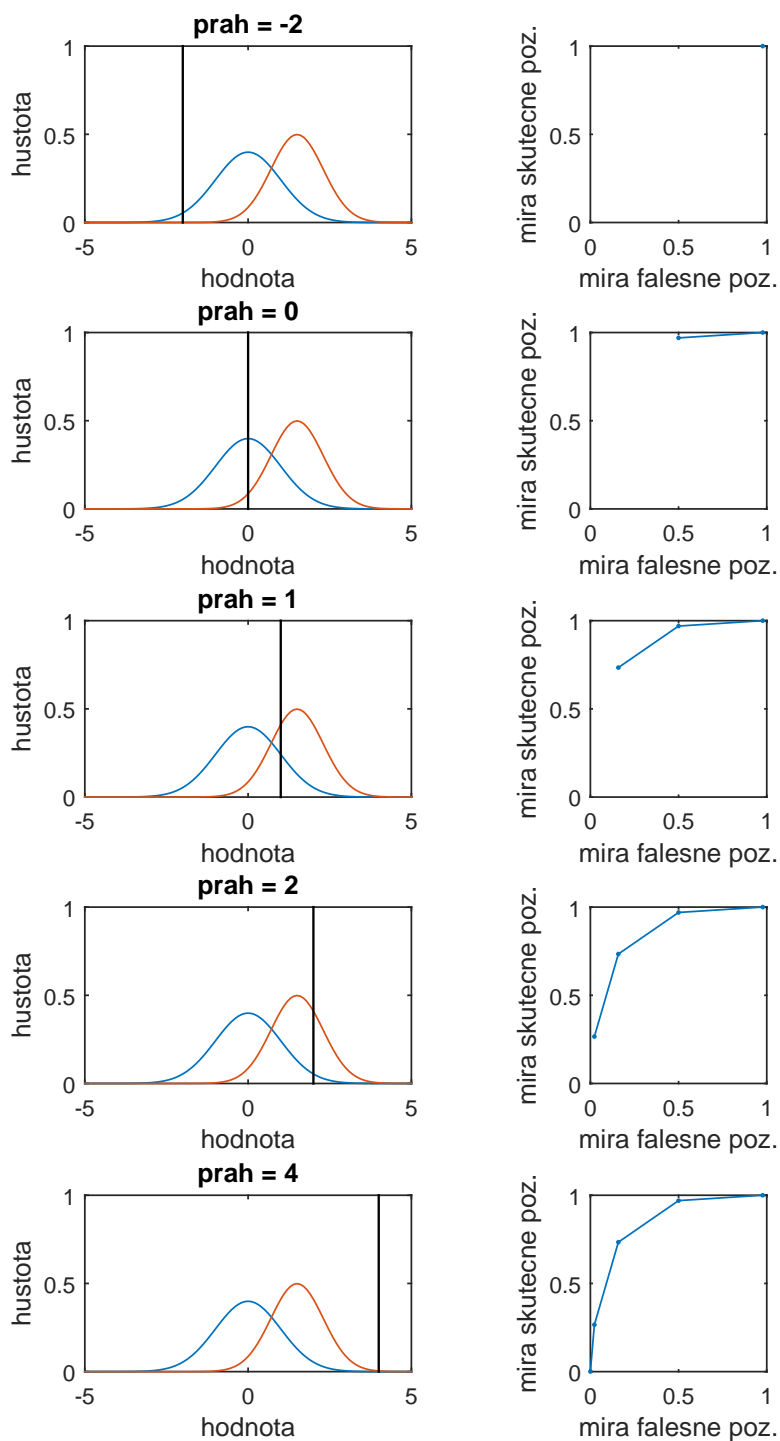
AUC	Area Under Curve – plocha pod křivkou
FFT	Fast Furier Transformation – rychlá Furierova transformace
FN	False Negative – falešně negativní
FP	False positive – falešně pozitivní
FPR	False Positive Rate – míra falešně pozitivních
IZS	Integrovaný záchranný sbor
MDF	Module Difference Function
ROC	Receiver operating characteristic
RZS	Rychlá záchranná služba
SNR	Signal to Noise Ratio – poměr signál/šum
TN	True Negative – skutečně negativní
TNR	True Negative Rate– míra správně negativních
TP	True Positive – skutečně pozitivní
WAV	Waveform Audio format – formát zvukových dat v PC
dB	decibel – jednotka poměrové logaritmické míry
dBFS	decibel Full Scale – jednotka logaritmické míry se vztažnou úrovní danou maximální hodnotou D/A či A/D převodníku
s.wav	zvukový soubor pouze sirény
sw.wav	zvukový soubor sirény s bílým šumem -20 dBFS
st.wav	zvukový soubor sirény s dopravním ruchem
stw.wav	zvukový soubor sirény s dopravním ruchem a bílým šumem -20 dBFS
sh.wav	zvukový soubor se silným dopravním ruchem
shw.wav	zvukový soubor se silným dopravním ruchem a bílým šumem -20 dBFS
sv.wav	zvukový soubor s velmi silným dopravním ruchem

svw.wav	zvukový soubor s velmi silným dopravním ruchem a bílým šumem -20 dBFS
sw1.wav	zvukový soubor sirény s bílým šumem o úrovni -20 dBFS
sw2.wav	zvukový soubor sirény s bílým šumem o úrovni -10 dBFS
sw3.wav	zvukový soubor sirény s bílým šumem o úrovni -8 dBFS
sw4.wav	zvukový soubor sirény s bílým šumem o úrovni -6 dBFS
sw5.wav	zvukový soubor sirény s bílým šumem o úrovni -3 dBFS
sw6.wav	zvukový soubor sirény s bílým šumem o úrovni -1 dBFS

SEZNAM PŘÍLOH

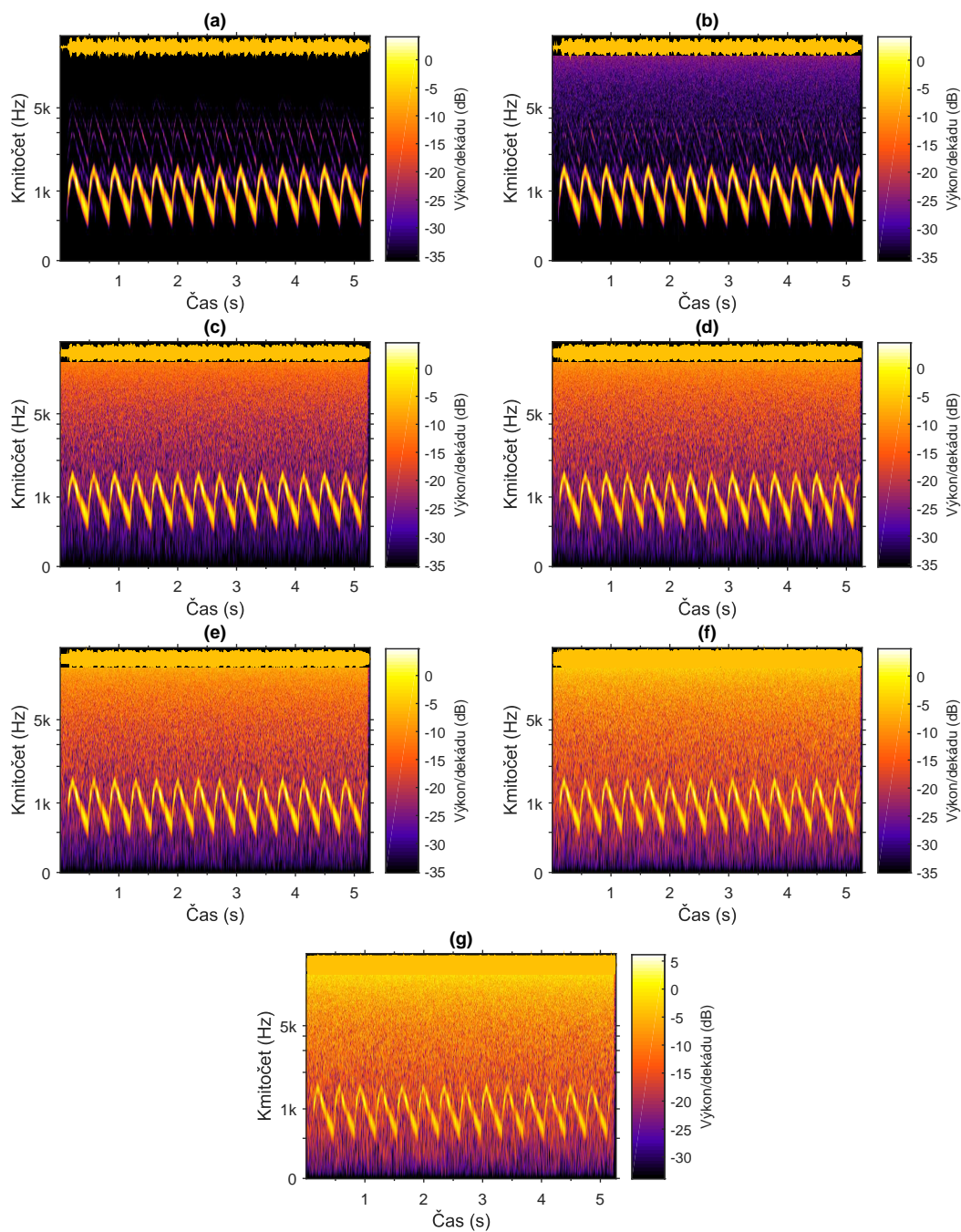
A Výpočet ROC	31
B Spektrogramy sirény se šumem	32
C Spektrogramy první sirény	33
D Spektrogramy první sirény s bílým šumem	34
E Spektrogramy druhé sirény	35
F Spektrogramy třetí sirény	36
G Spektrogramy čtvrté sirény	37
H Spektrogramy páté sirény	38
I Obsah souboru analyza.m	39
J Obsah souboru roc.m	40
K Obsah souboru rocvyk.m	41

A VÝPOČET ROC



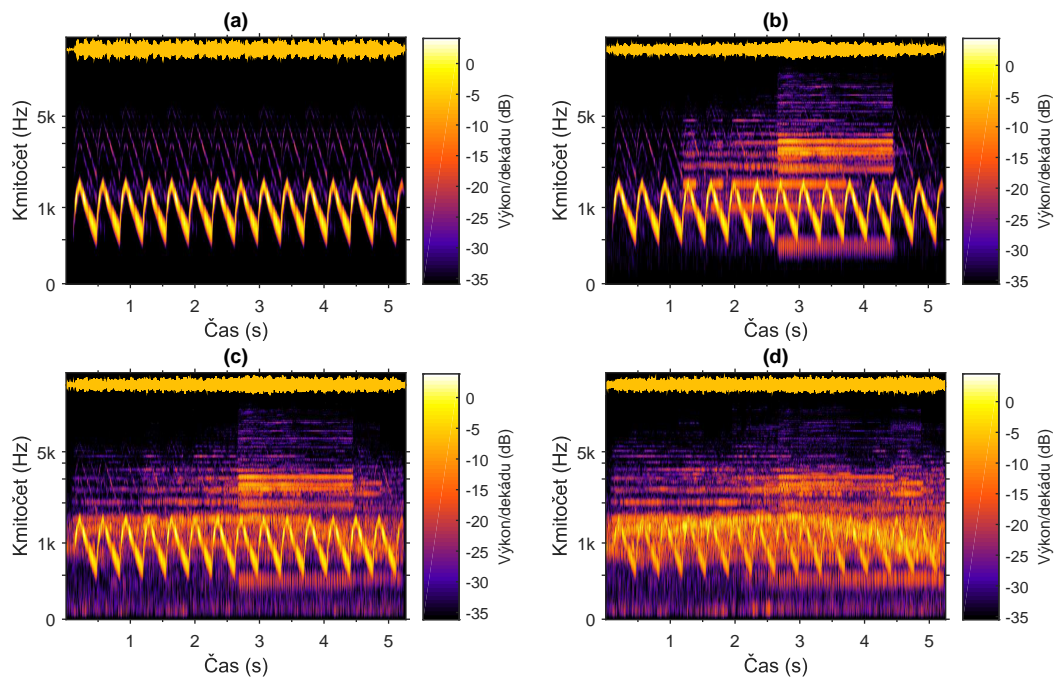
Obr. A.1: Výpočet a vykreslení ROC křivky

B SPEKTROGRAMY SIRÉNY SE ŠUMEM



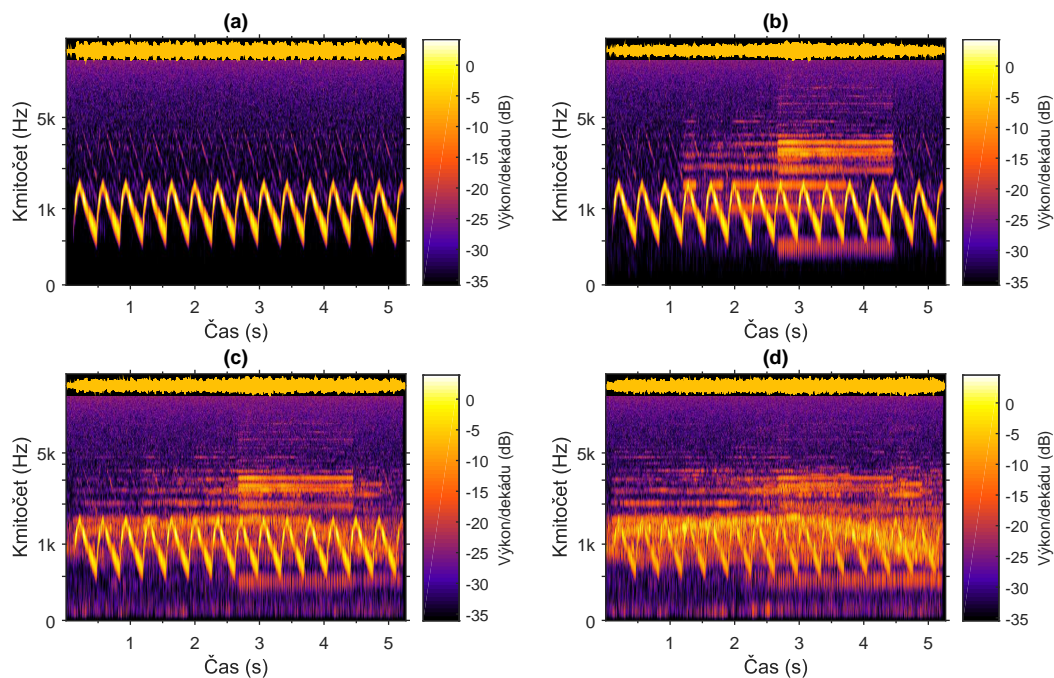
Obr. B.1: Spektrogramy zvuku první sirény s přidaným bílým šumem. (a) bez šumu; (b) bílý šum -20 dBFS; (c) bílý šum -10 dBFS; (d) bílý šum -8 dBFS; (e) bílý šum -6 dBFS; (f) bílý šum -3 dBFS; (g) bílý šum -1 dBFS

C SPEKTROGRAMY PRVNÍ SIRÉNY



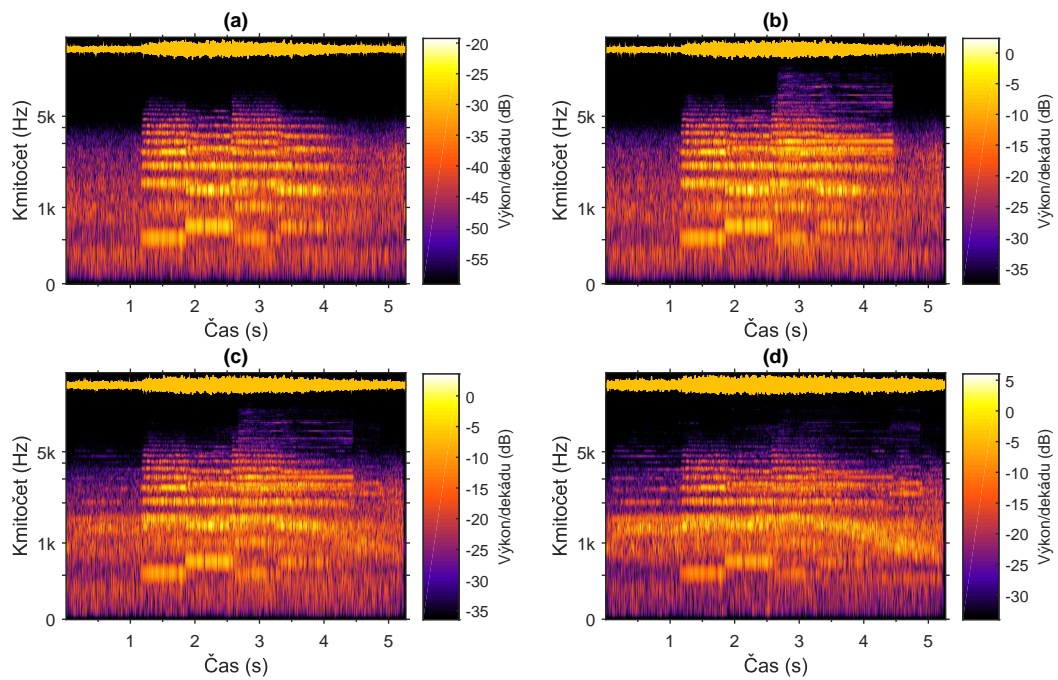
Obr. C.1: Spektrogramy první sirény s dopravním ruchem. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s doravním ruchem; (d) s velký dopravním ruchem.

D SPEKTROGRAMY PRVNÍ SIRÉNY S BÍLÝM ŠUMEM



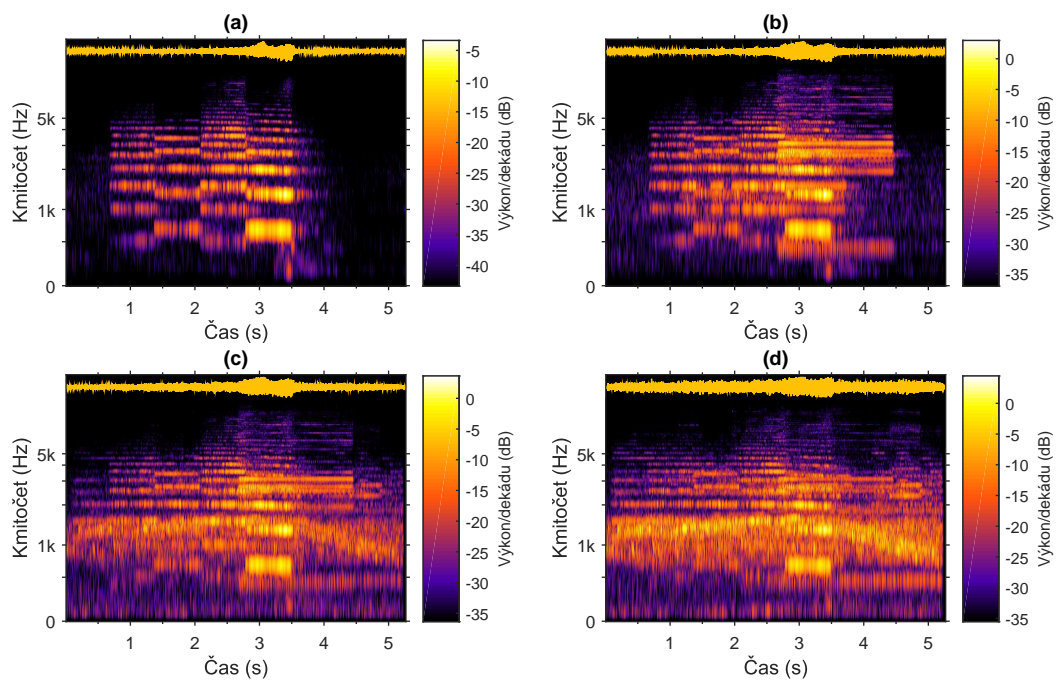
Obr. D.1: Spektrogramy čistého zvuku sirény a tří zarušených s přidaným bílým šumem o úrovni -20 dBFS. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s doravním ruchem; (d) s velkým dopravním ruchem.

E SPEKTROGRAMY DRUHÉ SIRÉNY



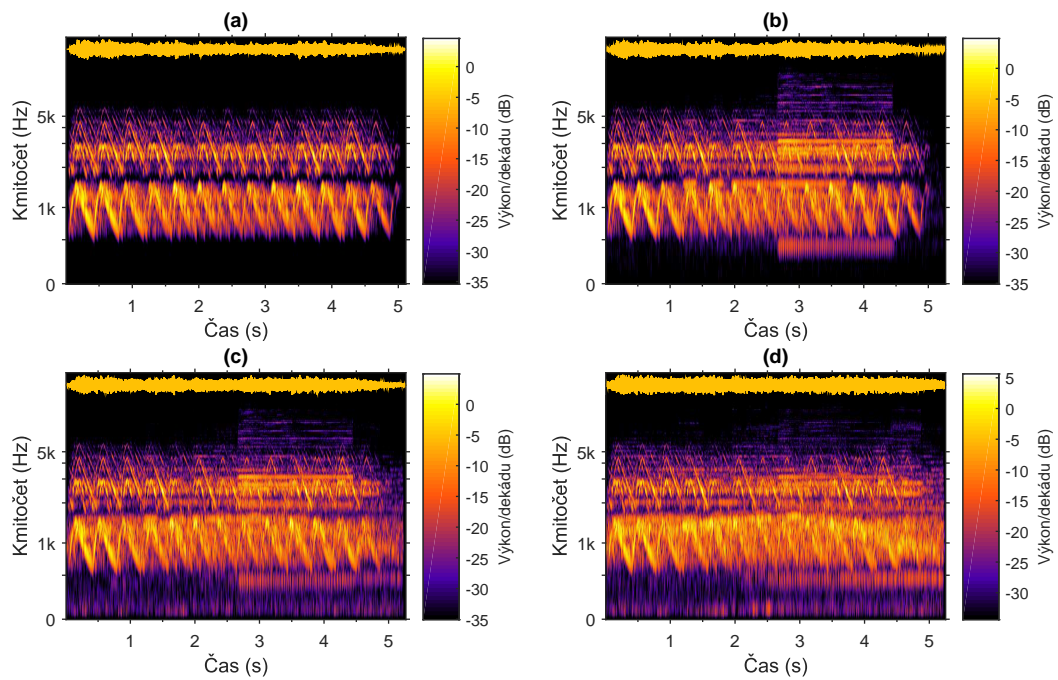
Obr. E.1: Spektrogramy druhé sirény s dopravním ruchem. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s dopravním ruchem; (d) s velký dopravním ruchem.

F SPEKTROGRAMY TŘETÍ SIRÉNY



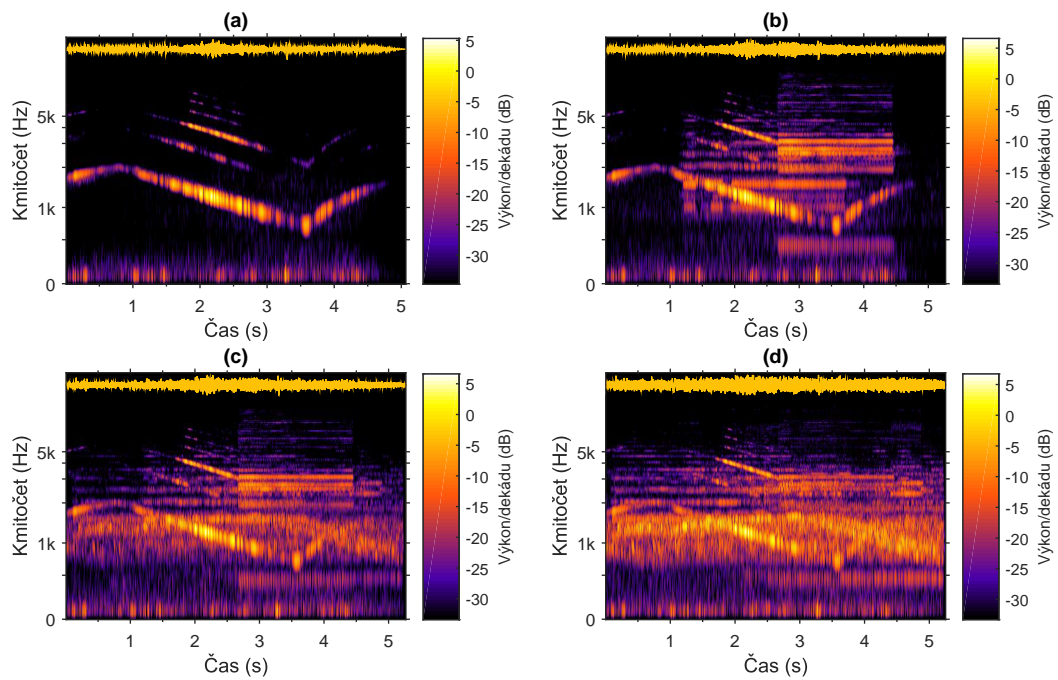
Obr. F.1: Spektrogramy třetí sirény s dopravním ruchem. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s doravním ruchem; (d) s velký dopravním ruchem.

G SPEKTROGRAMY ČTVRTÉ SIRÉNY



Obr. G.1: Spektrogramy čtvrté sirény s dopravním ruchem. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s doravním ruchem; (d) s velký dopravním ruchem.

H SPEKTROGRAMY PÁTÉ SIRÉNY



Obr. H.1: Spektrogramy páté sirény s dopravním ruchem. (a) bez dopravního ruchu; (b) s ruchem dopravní zácpy; (c) s doravním ruchem; (d) s velký dopravním ruchem.

I OBSAH SOUBORU ANALYZA.M

```
function [FFTmax]=analyza(file,lsample)
%% nahrani zvukoveho souboru, jeho rozkouskovani a FFT
% vyuziva toolbox Voicebox

%% cteni zvukoveho souboru

[data,fs]=readwav(file);          %nacteni celeho zvukoveho souboru
    % file - nazev souboru
    % data - vystupni vektor vzorku souboru
    % fs - vzorkovaci kmitocet

%% rozdeleni zvukoveho souboru na casti o delce 'lsample'
for d = 1:(length(data)/lsample)
    [data2]=readwav(file, '',lsample,(d-1)*lsample);
    FFTmax(d)=max(abs(rfft(data2)));
end
```

J OBSAH SOUBORU ROC.M

```
function [Pd,Pfa]=roc(data1,data2)
%funkce pro vypocet a vykresleni ROC krivky
%autor: Jitka Pomenkova
%data1=lsig_1;...signal se sumem
%data2=lsum_1;...sum

%=====cyklus pro vypocet hodnot Pd a ...
    Pfa=====
K1=sum(data1)/length(data1);
krok=K1/100;
hodnoty=0:krok:K1*3;
for j=1:length(hodnoty)
    K=hodnoty(j);
    P(j,1)=K;

%-----SIGNAL+SUM:sumaksig2
%pocitam dobrou detekci
%data1
T=0;F=0;
    for i=1:length(data1)
        if data1(i)>K % nasla jsem signal
            B(1,i)=1; T=T+1;
        else B(1,i)=0; F=F+1;
        end
    end
    Pd(1,j)=T/(T+F);

%-----SUM:sumak2
%pocitam falesnou detekci
%data2
Tt=0;Ft=0;
    for i=1:length(data2)
        if data2(i)>K %netrefila jsem se (nasla jsem signal), tj. ...
            opravdu falesna detekce
                A(1,i)=1; Tt=Tt+1;
        else A(1,i)=0; Ft=Ft+1;
        end
    end
    Pfa(1,j)=Tt/(Tt+Ft); %tj. pst, kdy jsem se netrefila v ...
        identifikaci sumu

end
plot(Pfa,Pd,'-','Marker','.')
```

K OBSAH SOUBORU ROCVYK.M

```
%close all
xlab='FPR';
ylab='TPR';

lsample = 700;
grx=3;
gry=5;

volba=1;
% moznosti:
% 1 - police1.wav
% 2 - police2.wav
% 3 - police3.wav
% 4 - siren.wav
% 5 - copcar.wav

%% nacteni souboru
if volba == 1
    disp('..nactani zvukovych souboru POLICE1')
    disp('..nactani sireny+ruch')
    respol = analyza('D:\BP\zvuky\police1.wav',lsample);
    respolb = analyza('D:\BP\zvuky\police+vse.wav',lsample);
    respolv = analyza('D:\BP\zvuky\police+vse2.wav',lsample);
    respolt = analyza('D:\BP\zvuky\police+traf.wav',lsample);

    disp('..nactani sireny+ruch+sum')
    respolts = analyza('D:\BP\zvuky\police+traf+sum.wav',lsample);
    respolbs = analyza('D:\BP\zvuky\police+vse+sum.wav',lsample);
    respolvs = analyza('D:\BP\zvuky\police+vse2+sum.wav',lsample);

    disp('..nactani ruchu')
    resnic = analyza('D:\BP\zvuky\ticho.wav',lsample);
    resback = analyza('D:\BP\zvuky\vse.wav',lsample);
    restraf = analyza('D:\BP\zvuky\trafficyam1.wav',lsample);
    resvse = analyza('D:\BP\zvuky\vse2.wav',lsample);

    disp('..nactani ruchu+sum')
    restsum = analyza('D:\BP\zvuky\traf+sum.wav',lsample);
    resbsum = analyza('D:\BP\zvuky\vse+sum.wav',lsample);
    resvsum = analyza('D:\BP\zvuky\vse2+sum.wav',lsample);

    disp('..nactani sireny+sum')
    respols = analyza('D:\BP\zvuky\police+sum.wav',lsample);
```

```

respols2 = analyza('D:\BP\zvuky\police+sum2.wav', lsample);
respols3 = analyza('D:\BP\zvuky\police+sum3.wav', lsample);
respols4 = analyza('D:\BP\zvuky\police+sum4.wav', lsample);
respols5 = analyza('D:\BP\zvuky\police+sum5.wav', lsample);
respols6 = analyza('D:\BP\zvuky\police+sum6.wav', lsample);

disp('..nacistani sumu')
ressum = analyza('D:\BP\zvuky\sum.wav', lsample);
ressum2 = analyza('D:\BP\zvuky\sum2.wav', lsample);
ressum3 = analyza('D:\BP\zvuky\sum3.wav', lsample);
ressum4 = analyza('D:\BP\zvuky\sum4.wav', lsample);
ressum5 = analyza('D:\BP\zvuky\sum5.wav', lsample);
ressum6 = analyza('D:\BP\zvuky\sum6.wav', lsample);

elseif volba == 2
disp('..nacistani zvukovych souboru POLICE2')
disp('..nacistani sireny+ruch')
respol = analyza('D:\BP\zvuky\police2.wav', lsample);
respolb = analyza('D:\BP\zvuky\pol2+vse.wav', lsample);
respolv = analyza('D:\BP\zvuky\pol2+vse2.wav', lsample);
respolt = analyza('D:\BP\zvuky\pol2+traf.wav', lsample);

disp('..nacistani sireny+ruch+sum')
respolts = analyza('D:\BP\zvuky\pol2+traf+sum.wav', lsample);
respolbs = analyza('D:\BP\zvuky\pol2+vse+sum.wav', lsample);
respolvs = analyza('D:\BP\zvuky\pol2+vse2+sum.wav', lsample);

disp('..nacistani ruchu')
resnic = analyza('D:\BP\zvuky\ticho.wav', lsample);
resback = analyza('D:\BP\zvuky\vse.wav', lsample);
restraf = analyza('D:\BP\zvuky\trafficejam1.wav', lsample);
resvse = analyza('D:\BP\zvuky\vse2.wav', lsample);

disp('..nacistani ruchu+sum')
restsum = analyza('D:\BP\zvuky\traf+sum.wav', lsample);
resbsum = analyza('D:\BP\zvuky\vse+sum.wav', lsample);
resvsum = analyza('D:\BP\zvuky\vse2+sum.wav', lsample);

disp('..nacistani siren\+sum')
respols = analyza('D:\BP\zvuky\pol2+sum.wav', lsample);
respols2 = analyza('D:\BP\zvuky\pol2+sum2.wav', lsample);
respols3 = analyza('D:\BP\zvuky\pol2+sum3.wav', lsample);
respols4 = analyza('D:\BP\zvuky\pol2+sum4.wav', lsample);
respols5 = analyza('D:\BP\zvuky\pol2+sum5.wav', lsample);
respols6 = analyza('D:\BP\zvuky\pol2+sum6.wav', lsample);

```

```

disp('..nacistani sumu')
ressum  = analyza('D:\BP\zvuky\sum.wav', lsample);
ressum2 = analyza('D:\BP\zvuky\sum2.wav', lsample);
ressum3 = analyza('D:\BP\zvuky\sum3.wav', lsample);
ressum4 = analyza('D:\BP\zvuky\sum4.wav', lsample);
ressum5 = analyza('D:\BP\zvuky\sum5.wav', lsample);
ressum6 = analyza('D:\BP\zvuky\sum6.wav', lsample);

elseif volba == 3
disp('..nacistani zvukovych souboru POLICE3')
disp('..nacistani sireny+ruch')
respol  = analyza('D:\BP\zvuky\police3.wav', lsample);
respolb = analyza('D:\BP\zvuky\pol3+vse.wav', lsample);
respolv = analyza('D:\BP\zvuky\pol3+vse2.wav', lsample);
respolt = analyza('D:\BP\zvuky\pol3+traf.wav', lsample);

disp('..nacistani sireny+ruch+sum')
respolts = analyza('D:\BP\zvuky\pol3+traf+sum.wav', lsample);
respolbs = analyza('D:\BP\zvuky\pol3+vse+sum.wav', lsample);
respolvs = analyza('D:\BP\zvuky\pol3+vse2+sum.wav', lsample);

disp('..nacistani ruchu')
resnic  = analyza('D:\BP\zvuky\ticho.wav', lsample);
resback = analyza('D:\BP\zvuky\vse.wav', lsample);
restraf = analyza('D:\BP\zvuky\trafficejam1.wav', lsample);
resvse  = analyza('D:\BP\zvuky\vse2.wav', lsample);

disp('..nacistani ruchu+sum')
restsum = analyza('D:\BP\zvuky\traf+sum.wav', lsample);
resbsum = analyza('D:\BP\zvuky\vse+sum.wav', lsample);
resvsum = analyza('D:\BP\zvuky\vse2+sum.wav', lsample);

disp('..nacistani sireny+sum')
respols = analyza('D:\BP\zvuky\pol3+sum.wav', lsample);
respols2 = analyza('D:\BP\zvuky\pol3+sum2.wav', lsample);
respols3 = analyza('D:\BP\zvuky\pol3+sum3.wav', lsample);
respols4 = analyza('D:\BP\zvuky\pol3+sum4.wav', lsample);
respols5 = analyza('D:\BP\zvuky\pol3+sum5.wav', lsample);
respols6 = analyza('D:\BP\zvuky\pol3+sum6.wav', lsample);

disp('..nacistani sumu')
ressum  = analyza('D:\BP\zvuky\sum.wav', lsample);
ressum2 = analyza('D:\BP\zvuky\sum2.wav', lsample);
ressum3 = analyza('D:\BP\zvuky\sum3.wav', lsample);

```

```

ressum4 = analyza('D:\BP\zvuky\sum4.wav', lsample);
ressum5 = analyza('D:\BP\zvuky\sum5.wav', lsample);
ressum6 = analyza('D:\BP\zvuky\sum6.wav', lsample);

elseif volba == 4
    disp('..nacistani zvukovych souboru SIREN')
    disp('..nacistani sireny+ruch')
    respol = analyza('D:\BP\zvuky\siren.wav', lsample);
    respolb = analyza('D:\BP\zvuky\siren+vse.wav', lsample);
    respolv = analyza('D:\BP\zvuky\siren+vse2.wav', lsample);
    respoltr = analyza('D:\BP\zvuky\siren+traf.wav', lsample);

    disp('..nacistani sireny+ruch+sum')
    respolts = analyza('D:\BP\zvuky\siren+traf+sum.wav', lsample);
    respolbs = analyza('D:\BP\zvuky\siren+vse+sum.wav', lsample);
    respolvs = analyza('D:\BP\zvuky\siren+vse2+sum.wav', lsample);

    disp('..nacistani ruchu')
    resnic = analyza('D:\BP\zvuky\ticho.wav', lsample);
    resback = analyza('D:\BP\zvuky\vse.wav', lsample);
    restraf = analyza('D:\BP\zvuky\trafficejam1.wav', lsample);
    resvse = analyza('D:\BP\zvuky\vse2.wav', lsample);

    disp('..nacistani ruchu+sum')
    restsum = analyza('D:\BP\zvuky\traf+sum.wav', lsample);
    resbsum = analyza('D:\BP\zvuky\vse+sum.wav', lsample);
    resvsum = analyza('D:\BP\zvuky\vse2+sum.wav', lsample);

    disp('..nacistani sireny+sum')
    respols = analyza('D:\BP\zvuky\siren+sum.wav', lsample);
    respols2 = analyza('D:\BP\zvuky\siren+sum2.wav', lsample);
    respols3 = analyza('D:\BP\zvuky\siren+sum3.wav', lsample);
    respols4 = analyza('D:\BP\zvuky\siren+sum4.wav', lsample);
    respols5 = analyza('D:\BP\zvuky\siren+sum5.wav', lsample);
    respols6 = analyza('D:\BP\zvuky\siren+sum6.wav', lsample);

    disp('..nacistani sumu')
    ressum = analyza('D:\BP\zvuky\sum.wav', lsample);
    ressum2 = analyza('D:\BP\zvuky\sum2.wav', lsample);
    ressum3 = analyza('D:\BP\zvuky\sum3.wav', lsample);
    ressum4 = analyza('D:\BP\zvuky\sum4.wav', lsample);
    ressum5 = analyza('D:\BP\zvuky\sum5.wav', lsample);
    ressum6 = analyza('D:\BP\zvuky\sum6.wav', lsample);

```

```

elseif volba == 5
    disp('..nacistani souboru COPCAR')
    disp('..nacistani sireny+ruch')
    respol  = analyza('D:\BP\zvuky\copcar.wav', lsample);
    respolb = analyza('D:\BP\zvuky\cop+vse.wav', lsample);
    respolv = analyza('D:\BP\zvuky\cop+vse2.wav', lsample);
    respolt = analyza('D:\BP\zvuky\cop+traf.wav', lsample);

    disp('..nacistani sireny+ruch+sum')
    respolts = analyza('D:\BP\zvuky\cop+traf+sum.wav', lsample);
    respolbs = analyza('D:\BP\zvuky\cop+vse+sum.wav', lsample);
    respolvs = analyza('D:\BP\zvuky\cop+vse2+sum.wav', lsample);

    disp('..nacistani ruchu')
    resnic  = analyza('D:\BP\zvuky\ticho.wav', lsample);
    resback = analyza('D:\BP\zvuky\vse.wav', lsample);
    restraf = analyza('D:\BP\zvuky\trafficejam1.wav', lsample);
    resvse  = analyza('D:\BP\zvuky\vse2.wav', lsample);

    disp('..nacistani ruchu+sum')
    restsum = analyza('D:\BP\zvuky\traf+sum.wav', lsample);
    resbsum = analyza('D:\BP\zvuky\vse+sum.wav', lsample);
    resvsum = analyza('D:\BP\zvuky\vse2+sum.wav', lsample);

    disp('..nacistani sireny+sum')
    respols = analyza('D:\BP\zvuky\cop+sum.wav', lsample);
    respols2 = analyza('D:\BP\zvuky\cop+sum2.wav', lsample);
    respols3 = analyza('D:\BP\zvuky\cop+sum3.wav', lsample);
    respols4 = analyza('D:\BP\zvuky\cop+sum4.wav', lsample);
    respols5 = analyza('D:\BP\zvuky\cop+sum5.wav', lsample);
    respols6 = analyza('D:\BP\zvuky\cop+sum6.wav', lsample);

    disp('..nacistani sumu')
    ressum  = analyza('D:\BP\zvuky\sum.wav', lsample);
    ressum2 = analyza('D:\BP\zvuky\sum2.wav', lsample);
    ressum3 = analyza('D:\BP\zvuky\sum3.wav', lsample);
    ressum4 = analyza('D:\BP\zvuky\sum4.wav', lsample);
    ressum5 = analyza('D:\BP\zvuky\sum5.wav', lsample);
    ressum6 = analyza('D:\BP\zvuky\sum6.wav', lsample);
end

%% vykresleni samostatnych ROC
figure(1)
disp('..vykresleni samostatnych ROC krivek')
disp('..sirena')
subplot(gry, grx, 1)

```



```

[Pd,Pfa]=roc(respol, resnic);
AUC(1)=0;
for a = 1:length(Pfa)-1
    AUC(1)=AUC(1)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
end
title('sirena')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'sirena v tichu')
text(0.1, 0.2, ['AUC= ', num2str(AUC(1))])

disp('..sirena + traf')
subplot(gry, grx, 4)
[Pd,Pfa]=roc(respol, restraf);
AUC(2)=0;
for a = 1:length(Pfa)-1
    AUC(2)=AUC(2)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
end
title('sirena+traf')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'pozadi dopravni zacky')
text(0.1, 0.2, ['AUC= ', num2str(AUC(2))])

disp('..sirena + vse')
subplot(gry, grx, 7)
[Pd,Pfa]=roc(respolb, resback);
AUC(3)=0;
for a = 1:length(Pfa)-1
    AUC(3)=AUC(3)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
end
title('sirena+vse')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'rusne pozadi')
text(0.1, 0.2, ['AUC= ', num2str(AUC(3))])

disp('..sirena + vse2')
subplot(gry, grx, 10)
[Pd,Pfa]=roc(respolv, resvse);
AUC(4)=0;
for a = 1:length(Pfa)-1
    AUC(4)=AUC(4)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);

```

```

end
title('sirena+vse2')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'velmi rusne pozadi')
text(0.1, 0.2, ['AUC= ', num2str(AUC(4))])

disp('..sirena + sum')
subplot(gry, grx, 2)
[Pd, Pfa]=roc(respol5, ressum);
AUC(5)=0;
for a = 1:length(Pfa)-1
    AUC(5)=AUC(5) + (Pfa(a) - Pfa(a+1)) * ((Pd(a) + Pd(a+1)) / 2);
end
title('sirena+sum')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'sirena s bilym sumem')
text(0.1, 0.2, ['AUC= ', num2str(AUC(5))])

disp('..sirena + traf + sum')
subplot(gry, grx, 5)
[Pd, Pfa]=roc(respolts, restsum);
AUC(6)=0;
for a = 1:length(Pfa)-1
    AUC(6)=AUC(6) + (Pfa(a) - Pfa(a+1)) * ((Pd(a) + Pd(a+1)) / 2);
end
title('sirena+traf+sum')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'zacpa s sumem')
text(0.1, 0.2, ['AUC= ', num2str(AUC(6))])

disp('..sirena + vse + sum')
subplot(gry, grx, 8)
[Pd, Pfa]=roc(respolbs, resbsum);
AUC(7)=0;
for a = 1:length(Pfa)-1
    AUC(7)=AUC(7) + (Pfa(a) - Pfa(a+1)) * ((Pd(a) + Pd(a+1)) / 2);
end
title('sirena+vse+sum')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'rusne s sumem')

```

```

text(0.1, 0.2, ['AUC= ', num2str(AUC(7))])

disp('..sirena + vse2 + sum')
subplot(gry, grx, 11)
axis square
[Pd, Pfa]=roc(respolvs, resvsum);
AUC(8)=0;
for a = 1:length(Pfa)-1
    AUC(8)=AUC(8)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
end
title('sirena+vse2+sum')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.1, 'velmi rusne s sumem')
text(0.1, 0.2, ['AUC= ', num2str(AUC(8))])

%% vykresleni vice ROC

disp('..krivky s ruchy')
subplot(gry, grx, 13)
roc(respol, resnic);
hold on
roc(respol, restraf);
roc(respolb, resback);
roc(respolv, resvse);
hold off
title('sum prostredi')
xlabel(xlab)
ylabel(ylab)

disp('..krivky s bilym sumem')
subplot(gry, grx, 14)
roc(respol, ressum);
hold on
roc(respolts, restsum);
roc(respolbs, resbsum);
roc(respolvs, resvsum);
hold off
title('bily sum')
xlabel(xlab)
ylabel(ylab)

%% dvojice ROC

disp('..dvojice sireny / + sum')

```

```

subplot(gry,grx,3)
roc(respol, resnic);
hold on
roc(respols, ressum);
hold off
title('sirena & sirena v sumu')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.2, ['dAUC= ', num2str(AUC(1)-AUC(5), '%7.5f')])

disp('..dvojice sirena + traf / + sum')
subplot(gry,grx,6)
roc(respol, restraf);
hold on
roc(respolts, restsum);
hold off
title('traffic & traffic v sumu')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.2, ['dAUC= ', num2str(AUC(2)-AUC(6), '%7.5f')])

disp('..dvojice sirena + vse / + sum')
subplot(gry,grx,9)
roc(respolb, resback);
hold on
roc(respolbs, resbsum);
hold off
title('ruch & ruch v sumu')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.2, ['dAUC= ', num2str(AUC(3)-AUC(7), '%7.5f')])

disp('..dvojice sirena + vse2 / + sum')
subplot(gry,grx,12)
roc(respolv, resvse);
hold on
roc(respolvs, resvsum);
hold off
title('velky ruch & velky ruch v sumu')
xlabel(xlab)
ylabel(ylab)
text(0.1, 0.2, ['dAUC= ', num2str(AUC(4)-AUC(8), '%7.5f')])

%% titulek velkeho grafu

if volba==1

```

```

        suplabel('Police1.wav','t')
elseif volba==2
        suplabel('Police2.wav','t')
elseif volba==3
        suplabel('Police3.wav','t')
elseif volba==4
        suplabel('siren.wav','t')
elseif volba==5
        suplabel('copcar.wav','t')
end

%% limity ROC
figure(4)
disp('.limity ROC')
figure(2)
set(gca,'XLim',[0,1]);

subplot(1,3,1)
plot([0 0 1], [0 1 1], 'LineWidth', 2);
title('(a)')
xlabel(xlab)
ylabel(ylab)
axis ('square')

subplot(1,3,2)
plot([0 1], [0 1], 'LineWidth', 2);
title('(b)')
xlabel(xlab)
ylabel(ylab)
axis ('square')

subplot(1,3,3)
plot([0 1 1],[0 0 1], 'LineWidth', 2);
title('(c)')
xlabel(xlab)
ylabel(ylab)
axis ('square')

%% princip vypoctu AUC
figure(5)
disp('.princip obdelnikove metody')
figure
[Pd,Pfa]=roc(respolbs,resbsum);
AUC(7)=0;
for a = 1:length(Pfa)-1
        AUC(7)=AUC(7)+(Pfa(a)-Pfa(a+1))*((Pd(a)+Pd(a+1))/2);
end

```

```

line('XData', [Pfa(a) Pfa(a)], 'YData', [0 (Pd(a)+Pd(a+1))/2], ...
      'LineStyle', '-', 'LineWidth', 1, 'Color','r')
line('XData', [Pfa(a+1) Pfa(a+1)], 'YData', [0 ...
      (Pd(a)+Pd(a+1))/2], 'LineStyle', '-', 'LineWidth', 1, ...
      'Color','r')
line('XData', [Pfa(a) Pfa(a+1)], 'YData', [(Pd(a)+Pd(a+1))/2 ...
      (Pd(a)+Pd(a+1))/2], 'LineStyle', '-', 'LineWidth', 1, ...
      'Color','r')
end
%title('police+vse+sum.wav')
xlabel(xlab)
ylabel(ylab)

%% hardnoise
figure(6)
disp('.sloucene ROC krivky')
disp('..ROC jednoho ruchu a tehoz ruchu se sumem')
roc(respolb,resback);
hold on
roc(respolbs,resbsum);

hold off
title('hard traffich rush & /w white noise')
xlabel(xlab)
ylabel(ylab)
legend(['hard noise; AUC= ',num2str(AUC(3),4)], ['hard noise with ...
      white noise; AUC= ',num2str(AUC(7),4)], 'Location','southeast')
text(0.5, 0.2, ['dAUC= ',num2str(AUC(3)-AUC(7), '%7.5f')])

disp('..ROC ruchu s bilym sumem')
figure(7)
roc(respolb,ressum);
hold on
roc(respolts,restsum);
roc(respolbs,resbsum);
roc(respolvs,resvsum);
hold off
legend(['s.wav; AUC= ',num2str(AUC(5),4)], ['st.wav; AUC= ...
      ',num2str(AUC(6),4)], ['sh.wav; AUC= ...
      ',num2str(AUC(7),4)], ['sv.wav; AUC= ...
      ',num2str(AUC(8),4)], 'Location','southeast')
title('ROC k?ivky sireny s bilym sumem')
xlabel(xlab)
ylabel(ylab)

disp('..ROC jen ruchu bez bileho sumu')

```

```

figure(8)
roc(respol, resnic);
hold on
roc(respol, restraf);
roc(respolb, resback);
roc(respolv, resvse);
hold off
legend(['s.wav; AUC= ', num2str(AUC(1), 4)], ['st.wav; AUC= ...
      ', num2str(AUC(2), 4)], ['sh.wav; AUC= ...
      ', num2str(AUC(3), 4)], ['sv.wav; AUC= ...
      ', num2str(AUC(4), 4)], 'Location', 'southeast')
title('ROC krivky sireny bez bileho sumu')
xlabel(xlab)
ylabel(ylab)

disp('..ROC jen s bilym sumem')
figure(9)
roc(respol, resnic);
hold on
roc(respols, ressum);
roc(respols2, ressum2);
roc(respols3, ressum3);
roc(respols4, ressum4);
roc(respols5, ressum5);
roc(respols6, ressum6);
hold off
legend(['s.wav; AUC= ', num2str(AUC(1), 4)], ...
      ['sum1.wav; AUC= ', num2str(AUC(2), 4)], ...
      ['sum2.wav; AUC= ', num2str(AUC(3), 4)], ...
      ['sum3.wav; AUC= ', num2str(AUC(4), 4)], ...
      ['sum4.wav; AUC= ', num2str(AUC(5), 4)], ...
      ['sum5.wav; AUC= ', num2str(AUC(6), 4)], ...
      ['sum6.wav; AUC= ', num2str(AUC(7), 4)], 'Location', 'southeast')
title('ROC krivky sireny bez bileho sumu')
xlabel(xlab)
ylabel(ylab)

```